Multilingual Financial Text Summarisation

Nadhem Zmandar, MEng, MSc

Supervisors: Dr. Mo El-Haj and Prof. Paul Rayson
School of Computing and Communications
Lancaster University

A thesis submitted for the degree of
Doctor of Philosophy

2024

Multilingual Financial Text Summarisation
Nadhem Zmandar, MEng, MSc.
School of Computing and Communications, Lancaster University
A thesis submitted for the degree of *Doctor of Philosophy*. 2024.

Abstract

With the increasing growth in the number of public firms worldwide, the volume of financial disclosures and financial texts in different languages and forms is increasing sharply; therefore, the study of Natural Language Processing (NLP) methods that automatically summarise content has grown rapidly into a major research area.

Financial communication is a vital component of market transparency and constitutes a key element for investor's confidence and the credibility and quality of a financial marketplace. Public firms are obliged to communicate regularly with their shareholders. The financial communication policy of a listed company reflects the regulatory constraints related to going public as well as the willingness of executives to regularly communicate with financial market players in a transparent, professional and responsive fashion. Financial narratives are used by firms to communicate with their stakeholders (investors, shareholders, customers, employees, financial analysts, regulators, lenders, rating agencies, and suppliers). Using financial communications, stakeholders could assess how the company can create value.

This thesis explores the financial text summarisation task from different angles. The goal is the development of general and scalable algorithms that can jointly improve the state of the art (SOTA) of the tasks of financial text summarisation and compare different methodologies that combine quantitative and qualitative performance. The ability to extract key information from financial documents and generate summaries in multiple languages is crucial for financial professionals and organisations. However, current text summarisation methods cannot accurately identify and extract relevant information when applied to financial texts due to the domain-specific nature of the language, the differing structures of financial documents, the complexity of financial concepts and the lack of well-developed language resources and models. This study investigates how to adapt different transformer language models (general and domain-specific) and alternative unsupervised techniques to generate a coherent summary, then presents different ways to measure the performance by combining automatic and human evaluations, and finally it proposes several adversarial attacks and statistical methods to test the robustness of the results. The models in this thesis provide state-of-the-art performance on the multilingual financial summarisation task.

This research contributes to the field of NLP by demonstrating our approach's effectiveness in multilingual financial text summarisation and provides valuable insights for developing multilingual text summarisation systems. This thesis targets three languages: Arabic, French and English. It targets three financial reporting frameworks and three financial market cultures. It deals with three types of documents: long unstructured documents (English reports), medium structured reports (French reports) and financial newswires (Arabic). In addition, this thesis combines several novel contribution types: dataset creation, ontology labelling, benchmarks for financial summarisation systems, monitoring of the NLP training process and pretraining of novel language models to fill the lack of domain and language-specific language models. This thesis also presents a novel approach to automatically summarising long financial text in multiple languages. Using advanced pretrained transformers, our system can accurately identify and extract essential information from financial documents and generate extractive and abstractive summaries in various languages.

Keywords: financial text-to-text model, financial summarisation, Arabic NLP, French NLP, English NLP, NLP system monitoring, NLP system evaluation.

Acknowledgements

I am incredibly grateful to my supervisors, Dr. Mo El-Haj and Prof. Paul Rayson, for their invaluable support, guidance, continuous motivation and advice throughout the past three years. I have had the privilege of knowing you, and I am so fortunate to have you as PhD supervisors and mentors. Your role went beyond academic support. You supported me throughout the difficult times in the past three years. Meeting with you weekly has created a deep relationship with you.

A heartfelt thanks go to my father, mother and sisters, without whom I would not be where I am today. My family is the foundation of this work. Their encouragement and support were crucial during my PhD journey. I am grateful to them for their love, support, and belief in my ability to complete my PhD program.

I express my sincere gratitude for the generous funding I received for my PhD. My research was funded by Lancaster University via an excellence PhD studentship provided from the Faculty of Science and Technology and the School of Computing and Communications. This PhD would not have been possible without this opportunity. I would also like to extend my thanks to all UCREL NLP group members.

I would also like to acknowledge Lancaster University for providing the right conditions to perform my research by giving me access to the High-End Computing cluster.

I would also like to acknowledge my co-authors Tobias Daudert, Sina Ahmadi and Abhishek Singh. Working with them helped me accelerate my NLP learning curve during the first year of my Ph.D.

Finally, I would like to express my gratitude to my relatives, to the staff at the School of Computing and Communications and to all my friends.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is: 66053

Nadhem Zmandar

Contents

List of Tables

List of Figures

List of Algorithms

Listings

Chapter 1

Introduction

1.1 Overview

In today's fast-paced global economy, access to timely and accurate financial information is crucial for making informed investment decisions. Due to the ever-growing number of companies around the world, the amount of financial disclosures and documents written in various languages is increasing drastically. Incidentally, corporations generate a substantial volume of financial disclosures, encompassing both qualitative and quantitative formats. The amount of qualitative financial narratives has increased drastically during the last ten years (El-Haj et al., 2020c). Consequently, there is an increasing demand for automated tools to assist analysts and investors in quickly recognising key points and patterns in financial narratives (El-Haj et al., 2019). Investigating language analysis techniques that can automatically summarise this content and extract Key Performance Indicators has become a key area of research. As financial narratives flow across linguistic boundaries, the need for automated multilingual financial text summarisation becomes evident. Natural Language Processing (NLP) is increasingly used in finance and business to analyse unstructured textual data. This trend was accelerated with the emergence of language models, which showed a high quality generative ability.

Financial reporting and communication requirements have been extended dramatically in recent years, especially after the financial crisis in 2008. Financial communications and investor relation management are becoming crucial parts of the financial markets and the fund management industry. All listed companies in regulated markets are required by law to regularly communicate their financial activities to their stakeholders. They are required to publish their financial reports and several other financial narratives on a regular basis.

The exchange of information is essential for market transparency and sets the basis for people to take action and constitutes a key element for investor confidence and the credibility and quality of a financial marketplace. In the financial sphere, this guarantees clarity and helps boost investors' trust while showing the dependability and excellence of a financial marketplace (PwC France, 2019; PwC France, 2020;

PwC France, 2021; PwC France, 2022). Companies listed on a stock exchange have been taking advantage of the "Financial Communication: Framework and Practices" guide (PwC France, 2022) to help with their financial communication practices for a number of years.

Listed companies have a legal requirement to communicate regularly with their shareholders. The financial communication policy of a listed company reflects the regulatory constraints related to going public as well as the willingness of executives to regularly communicate with financial market players in a transparent, professional and responsive fashion. Generally speaking, financial narratives are used by firms to communicate with their stakeholders (investors, shareholders, customers, employees, financial analysts, regulators, lenders, rating agencies, and suppliers). Using financial communications, stakeholders could assess how the company is able to create value.

There are different ways in which firms can communicate with their stakeholders, and we have different kinds of financial narratives (e.g. Annual Financial Reports, Preliminary Earning Announcements, Earning Announcements, Conference Calls, Press Releases, corporate social responsibility reports, Risk Management Reports, Audit Reports and IPO Prospectus) (El-Haj et al., 2019a). In addition, financial news agencies and equity research companies are also required to release news regularly about the activity of listed companies. Such announcements are used by investors and hedge funds to fine-tune their investment signal generation process. The choice of the reporting language and the format of reports is made by the financial regulator in a specific market (El-Haj et al., 2019b; El-Haj et al., 2019b).

In the past, the volume and velocity of financial textual data were manageable enough to be manually analysed by teams of human expert analysts. But with the increasing growth of the volume of financial disclosures in different languages and forms, financial NLP research is growing quickly and rapidly becoming a major research area since financial summarisation is different from general summarisation task because financial reports are quite long, and come in unstructured format, contains very technical vocabulary and are very subjective, especially in the UK, where companies use a free structure for financial reports. This motivates the use of natural language processing techniques to get the most from this textual data. This creates a huge research interest in financial natural language processing. The creation of tools that can scan financial documents, detect their structure and summarise them could be a big research challenge and will enhance the work of banks, financial analysts, hedge funds and central banks. In addition, the development of custom financial language models that generate financial headlines automatically could enhance the work of financial news agencies all over the world.

In a nutshell, the advance of Natural Language Processing (NLP) has revolutionised the way we approach financial text analysis. However, the domain of multilingual financial text summarisation remains intricate due to the intricate language nuances, domain-specific jargon, and cultural contexts embedded within financial communications in different languages.

1.2 Research Questions

The major question being answered in this thesis is how can we improve methods and techniques for the summarisation of financial text, whether it is a very long document (UK annual reports), medium size report (French financial statements) or very short text (financial news articles). This major question breaks down into the following research questions:

Research Question 1 *How can we enhance financial report summarisation?*

- *1.1 What are the most effective methods for financial annual report summarisation, and how do they vary for different narrative types?*
- *1.2 What methods are suitable for summarising annual reports (long documents) and financial news (short documents)?*
- *1.3 How can pre-trained language models be adapted for very long documents, taking into consideration the memory limitations of many Large Language Models for handling very long documents?*
- *1.4 What are the specific aspects of financial text that affect the quality of text summarisation and that make it different from other types of documents (e.g. legal, healthcare)?*

Research Question 2 *How do we improve evaluation methods for text summarisation?*

- *2.1 What other NLP methods can we exploit to better model text similarity in order to improve the evaluation process?*
- *2.2 What can we learn about the different categories of evaluation metrics by testing their behaviour during the training process?*
- *2.3 How can we confirm quantitative NLP results by human evaluation and readability measure?*

Research Question 3 *How can we move beyond the current focus only on English language for financial text summarisation?*

- *3.1 How can we enhance multilingual financial NLP research?*
- *3.2 How portable are multilingual financial text summarisation methods from English to other languages?*
- *3.3 How can we boost financial NLP research in under-researched languages by pertaining and finetuning field-specific language models?*

1.3 Contributions

This thesis makes several contributions to the field of multilingual financial text summarisation. We propose different types of contributions, such as novel pre-trained models, the creation of multilingual corpora and the exploration of different evaluation techniques. Our contributions could be summarised as follows:

- **Benchmarking the use of Transformers and other techniques for UK Financial Narrative Long Document Summarisation.**
  We conducted a benchmarking study targeting the use of transformers for English financial narrative long document summarisation. This study aims to benchmark different summarisation methods based mainly on pre-trained transformers. The summaries were evaluated using different metrics (embeddings-based and n-gram based) on a dataset of annual reports published by firms listed on the London Stock Exchange. The choice of different evaluation metrics is motivated by the fact that some evaluation metrics do not reflect true summarisation ability (e.g. rouge). Through this study, we want to see if different variants of metrics behave differently on the same task. The study also proposes a robust statistical significance testing framework for the results generated by the different transformers.

- **An adversarial analysis experiment for English evaluation metric testing.**
  Presenting a set of adversarial attacks to measure the robustness of some automatic evaluation metrics with three different corruption tasks. Adversarial attacks are text perturbations designed to test the effectiveness of the metrics. Our experiments involved corrupting a set of summaries generated by a summariser system. The thesis presents different sources of noise using BERT mask-filling, word-dropping, and word permutation methods. The noise is injected using a uniform distribution to avoid creating a bias in the distribution of corrupted tokens.

- **Multilingual Financial Word Embeddings.**
  Trained Multilingual Financial Word Embeddings for Arabic, English and French: In order to properly represent domain-specific vocabulary, styles and meanings, the thesis creates a set of novel financial word embeddings for three languages: English, French and Arabic.

- **Statistical significance testing study for English and French financial report summarisation results**. This study enables us to determine whether the test results are statistically significant, to avoid false discoveries from evaluating NLP models and to make sure that the probability of making a false claim is minimal.

- **The training of a financial Arabic Text-To-Text language model.**
  The training of a financial Arabic language model which is designed for text

generation. It is trained using a text-to-text approach. The model is based on AraT5 (Nagoudi et al., 2022a), a pretrained Arabic text-to-text model. It consists of encoder and decoder parts. It is a financial Arabic model pretrained in an encoder-decoder manner. The thesis explains from scratch the steps required to pre-train or continue the training of a language model and the different performance tests that could be performed on this language model.

- **Arabic Financial Reports Corpus.**
  The selection, creation, conversion, and cleaning of an Arabic financial reports corpus. The thesis describes our approach to collecting and converting PDF reports using OCR solutions into text files and how to clean the text files. The task is challenging, as financial statements are not readily available or centralised in one location. The corpus includes several types of financial documents from different Middle Eastern markets: auditor reports, earning announcements, accounting documents, quarterly reports (Q1, Q2, Q3, Q4), annual reports and management board reports. The data focused on major stock exchanges in the Middle East to collect our corpus. Our data is collected from the following Arab markets: KSA (TASI, NOMU), UAE (Dubai, Abu Dhabi), Kuwait, Oman, Qatar and Bahrain. The corpus is constituted as a diverse set of documents from different sectors and covers several categories. We have over 35 categories in this corpus (E.g. financial services, banking, insurance, telecommunication, oil and gas, energy, real estate, and utilities).

- **French financial narrative summarisation dataset.**
  The creation of CoFiF Plus: the first French financial narrative summarisation dataset provides a comprehensive set of financial text written in French. The dataset was extracted from French financial reports of companies listed on the French stock exchange. It is composed of 1,703 reports from the most capitalised companies in France (Euronext Paris), covering a time frame from 1995 to 2021. It was labelled using a set of extractive heuristic rules followed by a manual validation process. The “chairman highlights” the “financial statements”, and the “future perspectives” sections are used as gold standard summaries.

- **Training monitoring study within French narrative summarisation.**
  A Training monitoring study within French narrative summarisation. This study monitors evaluation metrics’ evolution during the French Financial Narrative Summarisation task. It was performed on the CoFiF Plus dataset and benchmarks four summarisation techniques (monolingual seq2seq models, multilingual seq2seq models, encoder2encoder models, and encoder2decoder models). The study evaluates the performance of different summarisation models using a range of metrics. It monitors n-gram-based(rouge 1,2, L, Lsum), embeddings-based (Frugal score, Bleurt, Bertscore) and statistical-based metrics(Depth score, Bary score and infoLM score). This study shows that the choice of the model architecture can significantly impact the performance of

summarisation models and highlight the importance of carefully selecting and monitoring metrics when evaluating summarisation systems.

- **Unsupervised and heuristic rule-based summarisation algorithms.**
  Benchmarking of different unsupervised and heuristic rule-based summarisation algorithms for French financial statements summarisation. This experiment shows the robustness of unsupervised techniques to extract the essential parts in an annual report, especially if the dataset is well annotated. This opens the doors for more unsupervised summarisation with the introduction of the new reporting framework in France using the XBRL markup language.

- **A framework to prepare Abstractive Financial Summarisation labelled datasets.**
  A proposed ultimate guide on producing a human-labelled financial narrative summarisation dataset using two different frameworks. The first framework is a chatGPT-aided solution to create a gold summary. The gold standard summary is generated by asking different questions and then organising and combining them to get coherent sentences. The second framework consists in manually extracting key sections from the report and paraphrasing them using one of the leading commercial solutions or with a finetuned text-to-text model on a downstream task of sentence paraphrasing.

- **Co-organisation of the Financial summarisation shared task (2021 and 2022 editions).**
  Co-organiser of the financial summarisation shared tasks. The Financial Narrative Summarisation (FNS 2021 and FNS 2022) aimed to demonstrate the value and challenges of applying automatic text summarisation to financial text written in English, usually called financial narrative disclosures. The task was hosted on Eval.ai platform to automate the evaluation of submissions. We supported five participating teams and 11 system submissions in the first year. The 2022 edition of the Financial Narrative Summarisation Shared Task included UK, Greek and Spanish financial annual reports. For the financial summarisation shared task (FNS at FNP 2021), we configured the challenge on eval.ai platform[1], coded the evaluation script and configured the DevOps part of the Python worker that will do the queuing of submitted solutions and evaluate them against the gold summaries one by one using the rouge 2.0 package. We divided the challenge into two main parts: training and testing and managed releasing the results. Using Eval-AI was very useful and enabled us to automate the evaluation of the submissions and to use custom evaluation phases and protocols.

[1] https://eval.ai/web/challenges/challenge-page/1070/overview

1.4 Organisation of the Thesis

- **Chapter 1: Introduction.**
  The introduction motivates the need to summarise financial text in a big data era. The chapter also presented a list of research questions this thesis will try to answer. Furthermore, the chapter gives an overview of the main contributions of this thesis. Finally, the chapter provides a list of technologies and libraries used in this thesis.

- **Chapter 2: Background and Related Work.**
  The background chapter gives an overview of the Natural Language Processing field. It summarises the different types and techniques used for text summarisation. Further, it defines financial narratives and the differences between different reporting frameworks and motivates the need to summarise financial text. Further, the literature on the evolution from neural networks to transformers and large language models will be reviewed. The chapter finishes by presenting a non-exhaustive list of metrics and techniques that could be used to evaluate system summaries.

- **Chapter 3: Methodology.**
  This chapter outlines the methodology employed in the design of the research process explored in this thesis. This high-level methodology chapter aims to provide a high-level overview of the research process and the interconnections between chapters, serving as a reference point for the next chapters.

- **Chapter 4: Financial Word Embeddings.**
  The chapter presents the different steps required to train financial custom word embedding models for English and French.

- **Chapter 5: Trilingual Financial Corpus.**
  This chapter contains an extensive presentation and exploration of different corpora used for this thesis. First, it describes the English financial narrative summarisation dataset. Second, it presents an Arabic corpus of financial annual reports and an Arabic financial news summarisation dataset. Finally, it presents a French financial statement summarisation dataset. The chapter presents detailed statistics and plots describing the dataset for every corpus.

- **Chapter 6: Long Document Financial Extractive Summarisation: Case study on English Financial Annual Report Summarisation .**
  The English dataset presented in the previous chapter is explored to perform a benchmarking study for using transformers and other techniques for UK Financial Narrative Long Document Summarisation. The chapter starts by giving a detailed literature review of previous work about English financial narrative summarisation. Further, it details the different summarisation techniques and the experimental methodology used in this study. After that,

we present the results and analysis. Then, the chapter presents a statistical significance testing for the results, an adversarial analysis attack study and finally, an ethical consideration section.

- **Chapter 7: Abstractive Summarisation: Use case on Arabic financial news summarisation.**
  This chapter contains a review of abstractive text summarisation and text-to-text language models. In addition, the chapter describes the procedure used to train a text-to-text financial Arabic model and details some evaluation results of the model. The chapter ends with a deep discussion about the usefulness of training monolingual language models and about the environmental footprint of the training process.

- **Chapter 8: Medium Size Financial Document Summarisation Monitoring: A Case Study on French Companies' Annual Financial Report Summarisation.**
  The chapter presents an extensive literature review of NLP monitoring and evaluation metrics used in this French summarisation work. Then, the chapter presents a way to monitor the training of different evaluation metrics during the task of French narrative summarisation. It monitors n-gram-based(rouge 1,2, L, Lsum), embeddings-based (Frugal score, Bleurt, Bertscore) and statistical-based metrics(Depth score, Bary score and infoLM score), and benchmarks four summarisation techniques (monolingual seq2seq models, multilingual seq2seq models, encoder2encoder models, and encoder2decoder models). Further, the chapter compares some supervised models with a wide range of unsupervised techniques and reinforcement learning summarisers. It ends by performing a statistical significance analysis to validate the results.

- **Chapter 9: Conclusion.**
  The conclusion summarises the thesis, revisits the research questions, details some limitations of the work and opens the doors to future work related to multilingual financial summarisation.

1.5 Motivation and Practical Use Cases for Financial Narrative Summarisation

Financial narratives are becoming predominant in the life cycle of financial markets. In fact, nowadays, investors read the news when buying stocks and do a deep due diligence process when acquiring a firm. They also evaluate a firm's potential growth and future cashflows by reading and analysing its annual report. The amount of available narratives is increasing continuously. This is motivated by the growth of the number of listed firms worldwide. The main challenge in the world of financial data is that we are moving from the era of structured quantitative data to the era of unstructured qualitative data due to the emergence of alternative financial data and the increasing number of listed companies that generate a massive amount of financial narratives. In this section we will highlight some useful use cases where companies could benefit from financial text summarisation in different fields of applications.

- Finance & investments
  - Quantitative or systematic trading hedge funds generate investment signals automatically using technical indicators or sentimental analysis. Hedge funds get financial news from their proprietary data providers. An automatic financial news summariser would play a crucial role in building a real-time sentimental analysis signal investment engine.
  - Retail investors and financial news readers are overwhelmed with different data sources. The current investment and financial news applications scrape articles from different magazines and organise links. A financial magazine reader will seek a financial news summarisation tool for one or multiple articles in order to have daily insights about financial markets from one source. Such a service could be an over-the-top service for apps such as investing.com.
  - NLP can automatically synthesise and summarise the SEC fillings and extract signals around sentiments and targeted events. Analysts can easily navigate transcripts and view the clustering of positive and negative sentiment or trends over time and across one or multiple companies to uncover themes that are otherwise missed in traditional analysis. A paragraph summariser would help the analyst cope with this analysis and go straight forward to the right place in the financial report.
  - Banking: We can use text summarisation as a step towards credit scoring and risk assessment. Fintechs are developing new models that extract the worthiness of borrowing for a country or a firm from the rating agency reports. Summarising these reports could accelerate this process.
- Insurance:
  - The insurance sector contributes the most to the GDP of developed countries. These firms generate thousands of financial documents yearly. All of those companies have an investment subsidiary. In such context, financial text summarisation would be a very interesting application case. Insurance firms must summarise regulatory documents, news and other sources to generate meaningful insights.
- Corporate:
  - Accounting and auditing firms are witnessing an urgent need for algorithms that manipulate financial reports. Computational linguistics is essential in the auditing and accounting profession that is willing to expand the new

AI & Big data techniques to deliver higher quality auditing and consulting services and transform financial reporting.

- Text summarisation would allow corporates to view thousands of financial reports, financial communications and financial news automatically at scale and draw actionable insights from any source of text.
- Financial consulting firms (Rothschild, Jefferies, liberty) and investment banks (JP Morgan, Morgan Stanley) need to do a due diligence (long process) before executing large Mergers and Acquisition transactions. So, their financial analysts will be obliged to read thousands of documents. An automatic financial text summariser would help them gain time and energy and satisfy their clients rapidly.

1.6 Technologies and Libraries

The following libraries and software have helped to prepare this thesis:

- Gensim (Rehurek et al., 2011) is an open-source library for unsupervised topic modelling, document indexing, and other natural language processing functionalities using statistical machine learning.

- Google Colab [2]: is a hosted Jupyter Notebook service that gives an environment ready for machine learning and data science research. In addition, it provides free access to computing resources, including GPUs and TPUs.

- Anaconda (*Anaconda Software Distribution* 2020) is an open-source software distribution Python and R programming languages used for data science and scientific computing.

- Spacy (Honnibal et al., 2017). is an open-source software library for advanced natural language processing written in Python and Cython.

- NLTK (Bird et al., 2009). Natural Language Toolkit is a Python library to manipulate natural language data.

- Pytorch (Paszke et al., 2017) is A Python library (created by Facebook) that handles essential steps like computing gradients, GPU acceleration, data parallelisation, executing hidden layers, optimisers, and loss functions.

- Tensorflow (Abadi et al., 2015) is a free and open-source software library system (developed by Google Brain) for machine learning and artificial intelligence. It is widely used to build neural networks.

[2] https://research.google.com/colaboratory/

- Wandb.ai (Biewald, 2020): Weights & Biases enables to monitor ML/NLP experiments, manage and version the datasets data using a centralised cloud server.

- Transformers (Wolf et al., 2020) is an open-source library that supports Transformer based architecture and gives easy access to pretrained language models hosted on huggingface cloud servers.

- Datasets (Lhoest et al., 2021). Is an open source library developed by the Huggingface team. It was developed to address the challenges of NLP dataset management and of NLP system evaluations.

- evaluate[3]: is a Python library that makes evaluating the performance of NLP models easier and more standardised.

- Accelerate (Sylvain Gugger, 2022) is an open-source library created by the HuggingFace team. It is created for PyTorch users to optimise the use of multi-GPUs/TPU/fp16.

- Python 3 (Van Rossum et al., 2009) Python is an important programming language for data science, Deep Learning and web development. It is an interpreted high-level language. It can be used in two approaches: Jupyter notebook (for research) and scripting (for production).

- R programming Language (R Core Team, 2016) is a language and environment for statistical computing. It was used for some statistical calculations for this thesis.

- BASH Shell (GNU, 2007) is a Unix shell and command language. It was mainly used to interact with the high-end computing cluster of Lancaster university in order to run training jobs.

- Pandas (Mckinney, 2011). A software Python library which is very useful for data manipulation and analysis.

- Matplotlib (Hunter, 2007). A stable plotting library used for creating different data visualisations in Python.

- Numpy (Oliphant, 2015) is the fundamental package for scientific computing in Python programming language.

- Scikit-learn (Pedregosa et al., 2011) is a free Simple and efficient software machine-learning library for Python. It is built on NumPy, SciPy, and matplotlib.

[3] `https://github.com/huggingface/evaluate`

- Scipy (Virtanen et al., 2020) is a python library used for scientific computing and technical computing.
- seaborn (Waskom et al., 2017) is a Python data visualisation library built on matplotlib. It provides a high-level interface for drawing statistical graphics.
- Beautiful Soup (Richardson, 2007) is a Python library for parsing data out of HTML and XML files.
- Scrapy (Kouzis-Loukas, 2016) is an open-source Python web-crawling library.
- CUDA[4] (Cook, 2012): The CUDA Toolkit offers a comprehensive development environment for those who write in C or C++ and want to create applications that are accelerated by GPUs.

[4] https://developer.nvidia.com/cuda-toolkit

Chapter 2

Background and Related Work

2.1 Overview of Natural Language Processing

The field of Natural Language Processing lies at the intersection of computer science, artificial intelligence and computational linguistics. It aims to enable computers to understand, interpret and generate human language by developing algorithms and models that can analyse and interpret written or spoken words and sentences (Khurana et al., 2022; Cambria et al., 2014). The research in this field has been underway for many years but is fast-moving and constantly evolving. A recent significant shift occurred in 2017 with the emergence of large language models, and recently NLP gained higher public interest with the massive creation of generative AI solutions and due to its applications in several use cases such as chatbots (Khurana et al., 2022; Bahja, 2020; Jurafsky et al., 2024).

NLP tasks can be categorised based on the types of the task: e.g. discriminative tasks such as Part of Speech POS tagging (Chiche et al., 2022; Kumawat et al., 2015), text and token classification (Li et al., 2021; Dogra et al., 2022), named entity recognition (Mohit, 2014; Li et al., 2022a), or generative tasks such as text summarisation (Mridha et al., 2021; Cajueiro et al., 2023; Yadav et al., 2022), text generation (Iqbal et al., 2022; Li et al., 2022b; Fatima et al., 2022), machine translation (Wang et al., 2022; Jiang et al., 2020; Koehn, 2010), mask filling (Gao et al., 2023; Donahue et al., 2020; Song et al., 2019). In addition, there are several books that covered a deep explanation of natural language processing (Clark et al., 2010; Jurafsky et al., 2009b; Goldberg, 2017; Manning et al., 1999; Jurafsky et al., 2009a; Bird et al., 2009; Manaris, 1998). In a nutshell, Natural Language Processing is a wide research area that is rapidly evolving targeting more and more applications.

2.2 Background of Text Summarisation

There has been an increased interest in text summarisation research in the last decade, proving that research on text summarisation is still relevant. Research is distributed over different sub-categories: extractive, abstractive, single document, multi-document, domain, and real-time summarisation (Widyassari et al., 2022).

2.2.1 Historical Milestones of Text Summarisation

Recent innovations in deep learning and natural language processing have enabled the creation of more advanced text summarisation methods, like pre-trained language models such as BERT (Devlin et al., 2019) and GPT (Brown et al., 2020). By refining these models for summarisation tasks, outstanding results have been achieved on standard datasets. However, this would not be possible without the huge efforts of the NLP and summarisation research community over recent decades. Early work in text summarisation can be traced back to the 1950s and 1960s, when researchers began exploring methods for automatic abstracting of scientific articles. In the 1970s and 1980s, researchers continued to develop rule-based systems for text summarisation, which relied on manually crafted linguistic and semantic rules. After that, the research in the area of text summarisation has seen a great deal of progress over recent years (since the 1990s), with numerous strategies and techniques being tested to create summaries of texts that are accurate and comprehensive. Orasan (2019) presented a detailed report entitled "Automatic summarisation: 25 years On". This article presents a quick overview of the main developments and advances in the automatic summarisation field from the last 25 years. In addition, there are other survey articles that presented a deep overview about the historical milestones of text summarisation. We can cite few examples such as (Saggion et al., 2012; Lloret et al., 2011; Nenkova et al., 2012; Gupta et al., 2019).

Given the rapid developments in text summarisation, the different milestones could be summarised as follows:

1950s - Early Research: Early work on automatic text summarisation began with research in information retrieval and natural language processing. The focus was on extracting relevant sentences or phrases to create concise summaries. The initial methods involved extracting sentences based on their statistical properties or using keyword-based techniques.

1958 - The Luhn Algorithm: (Luhn, 1958) is one of the earliest approaches to automated text summarisation. Luhn (1958) proposed a frequency-based method for selecting important words and phrases from a document to create an extractive summary. This algorithm is one of the earliest and most important experiments in automated text summarisation.

1960s - (Edmundson, 1969a) presented "New methods in automatic extracting" where they introduced a heuristic-based method for sentence extraction in text summarisation.

1970s - Sentence Extraction: Research in the 1970s focused on methods to extract important sentences from a text and assemble them into a coherent summary. These approaches relied on heuristics rule based algorithms and linguistic patterns.

1990s - Machine Learning Approaches: In the 1990s, with the emergence of machine learning and statistical techniques for natural language processing, researchers started to investigate data-driven strategies for text summarisation using machine learning approaches. The adoption of machine learning techniques, particularly supervised learning, brought improvements to summarisation models. Researchers explored methods like decision trees, naive bayes and neural networks (Hovy et al., 1998; Kupiec et al., 1995). In addition, text summarisation techniques started incorporating linguistic and semantic analysis which allowed for more accurate and context-aware summaries. With these advances, researchers began to differentiate between two main approaches: abstractive and extractive summarisation. The latter got more interest with an accelerated development of extractive summarisation systems, which involves selecting important sentences or phrases from the original text to produce a summary.

The late 1990s saw the creation of first workshops dedicated to the topic of NLP and automatic text summarisation such as the Association of Computational Linguistics annual meeting ACL 97/EACL 97. Workshops dedicated to automatic text summarisation are now organised on a regular basis during major conferences such as ACL, EACL, NAACL, COLING, LREC and RANLP.

2000s - Automatic Evaluation Metrics: Over time, researchers have developed evaluation metrics to assess the quality of automatic summaries. The introduction of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics (Lin, 2004) provided the first standardised way to evaluate the quality of automatic summaries. ROUGE measures the overlap between generated summaries and human-written references. Rouge have become standard in the field, providing a benchmark for comparing different summarisation models. In the early 2000s, "TIPSTER SUMMAC Text Summarisation Evaluation", the first evaluation conference dedicated to automatic summarisation, was organised (Mani et al., 1999; Mani et al., 2002).

On another side, in the 2000s, researchers began to explore more advanced techniques for text summarisation, including abstractive summarisation, which involves generating new words and sentences that capture the essential meaning of the original text. These approaches often used neural network-based models, such as sequence-to-sequence models, to generate summaries.

2011 - Nenkova et al. (2011b) presented a survey entitled "Automatic summarisation: Foundations and Trends" providing an overview of automatic summarisation techniques, including extractive and abstractive methods.

2014/2015 - Introduction of Seq2Seq Models: The application of sequence-to-sequence (Seq2Seq) models, particularly using recurrent neural networks (RNNs) and later transformers, revolutionised abstractive summarisation. These models generated summaries by predicting words instead of extracting sentences. Rush et al. (2015) introduced an early application of neural attention models for abstractive summarisation.

2017 - Pointer-Generator Networks: The introduction of pointer-generator networks combined the benefits of extractive and abstractive summarisation approaches. This model can decide whether to copy words from the input text or generate new words (See et al., 2017).

2017 - Introduction of Attention Mechanism: The attention mechanism, popularised by the Transformer model architecture, revolutionised the text summarisation field. It allowed models to focus on relevant parts of the input during the process of summarisation, hence improving the quality of the generated summaries (Vaswani et al., 2017).

2019 - Pretrained Language Models: Pretrained large-scale language models such as BERT and GPT brought significant improvements to summarisation tasks. Fine-tuning these models on summarisation datasets has shown promising results in generating meaningful summaries and led to state-of-the-art (SOTA) results (Liu et al., 2019a).

2020s - Advanced Usecases, Techniques and Capacities: The research in the field of automatic summarisation has witnessed an acceleration since the last three years. Text summarisation has found practical applications in various domains. It is used for news summarisation, document summarisation and social media summarisation. In addition, there is a big interest in exploring reinforcement learning from human feedback (RLHF) and reinforcement learning with hybrid rewards to enhance the quality of generated summaries. Added to that, multilingualism in general and multilingual summarisation, where models can generate summaries in multiple languages, has gained great success. This has opened up opportunities for cross-lingual and multilingual content summarisation (Hasan et al., 2021; Grashchenkov et al., 2022; Xue et al., 2021; Cao et al., 2020). And recently, we had a surge in summarisation APIs[1] and tools. These tools enable users to summarise text with just a few lines of code or through user-friendly interfaces. Finally, we have the emergence of a set of newly introduced evaluation metrics based on language models and statistical measures. These metrics comes to complete the previously used n-gram-based metrics.

These milestones have shaped the landscape of text summarisation. Ongoing research and developments in natural language processing (NLP) and machine learning will continue to enhance the effectiveness and efficiency of text-automatic summarisation systems.

[1]`https://www.edenai.co/post/best-summarization-apis`

2.2.2 Different Types of Summarisation Techniques

The Figure 2.1 shows an overview of text summarisation sub-fields (based on the purpose, training algorithm, input type, summary type, summary language):

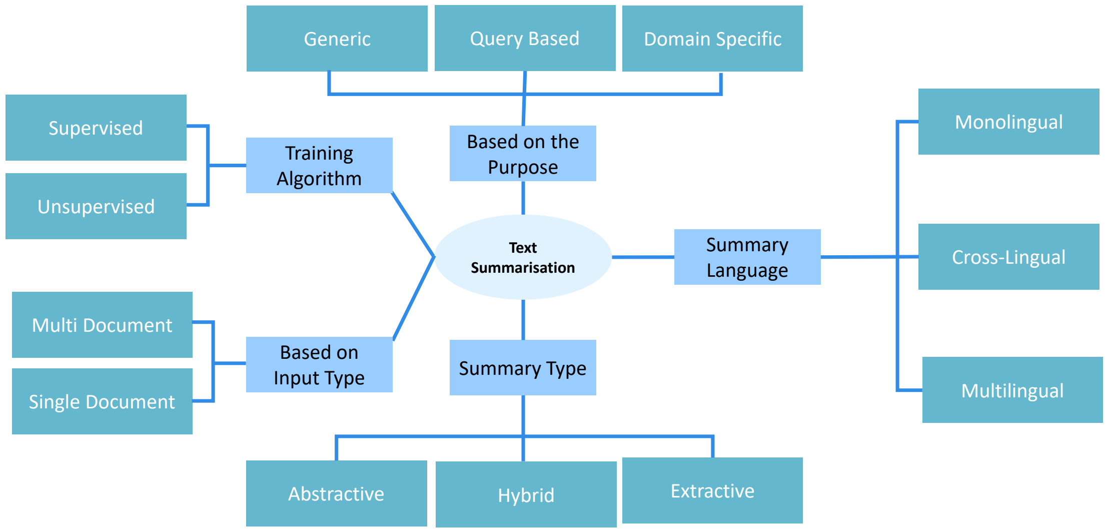


Figure 2.1: Overview of text summarisation: Illustration of different categories within text summarisation.

Summarisation approaches can also be classified as follows:

- Topic modelling (Celikyilmaz et al., 2010): In these techniques, the content or topical distribution of the final summary is predicted using a probabilistic distribution framework.
- Graph based summarisation (Paul et al., 2010; Erkan et al., 2004a; Mihalcea et al., 2004): which creates a graph where sentences are nodes and edges are similarities, then tries to find the most central sentences in a document's graph. They are often used as strong baselines or toplines.
- Supervised summarisation models (CHALI et al., 2012), where the selection of sentences in the system summary is learned using a supervised learning training framework.
- Heuristic rule-based (Carbonell et al., 1998; Lin et al., 2010): this summarisation approach works uses a greedy selection of the content based on a predefined rule-based algorithm.
- Neural networks: Before the emergence of transformers, there have been some efforts to apply neural networks (RNN-LSTM) and sequence-to-sequence

modelling (Sutskever et al., 2014a) for generating abstractive and extractive summaries of short and medium size texts (Rush et al., 2015).

- Setting a reward function and optimising the process to maximise it with respect to some constraints. The best example for such use case is reinforcement learning (Paulus et al., 2018).

- Statistical–based Summarisation: Many summarisation systems use statistical approaches to identify important sentences, such as Hidden Markov Models, Expectation Maximisation(Knight et al., 2000), Vector Space Model(Galanis et al., 2008; Berger et al., 2000) and the TFIDF (Term Frequency Inverse Document Frequency) (Salton et al., 1988).

- Cluster–based Summarisation: It is the act of assigning a group of observations into subgroups, known as clusters. Over the last years, clustering has been used for extractive summarisation (Dunlavy et al., 2007).

- Information Retrieval: Chowdhury (2003) developed a proficient method focused on Information Retrieval (IR) to extract sentences from text for summarisation.

2.2.3 Automatic Text Summarisation

The task of text summarisation is to condense long input documents into short and brief summaries, conveying the most important aspects while preserving the content, meaning and any relevant information. Summaries play a crucial role in communicating written information (Nenkova et al., 2011a; Zeng et al., 2016; Lierde et al., 2019; Shakhovska et al., 2019; Huang et al., 2020; Yao et al., 2015; Wang et al., 2016; Edmundson, 1969b). Summaries can be defined as a document reduced only to its essential content. A summary helps readers to grasp the material more easily and quickly. In a digital world with ever-increasing volumes of writing available online, summaries are playing a vital role in synthesising information into a digestible format for readers.

Text summarisation faces the challenge of generating a coherent and informationally accurate output while distilling key points. Furthermore, text summarisation faces the engineering challenge (related to model capacity) of the asymmetric lengths of inputs and outputs. Moreover, evaluating text summarisation is an interesting topic due to the subjectiveness of the process and the fact that there may be multiple valid summaries for each source document. Considering all these factors together, it becomes clear why automatic summarisation has significant research attention and became a recurrent area of study (Pasunuru et al., 2018; Narayan et al., 2017; Fan et al., 2018; Durrett et al., 2016; Tan et al., 2017).

In order to train a summarisation model, it is necessary to have a set of gold-standard summaries for each input text. These gold standard summaries can be obtained by extracting relevant information from the original text or by creating them through human expertise. During the training process of the NLP model, these gold standard summaries serve as the target label that we aim to achieve.

Hovy (2022) defines as follows: Extracts are summaries created by re-using portions (words, sentences, etc.) of the input text verbatim, while abstracts are created by re-generating the extracted content.

2.2.3.1 Extractive Summarisation

The approach of the extractive summarisation method directly selects and outputs the salient phrases in the original document (Jing et al., 1999; Knight et al., 2002). It focuses on extracting key sentences and arranging them in a manner to create a coherent summary that contains the relevant information. It is also possible to generate summaries based on paragraphs rather than sentences (Kågebäck et al., 2014; Steinberger et al., 2005; Gillick et al., 2009; Collins et al., 2017; Verma et al., 2017; Al-Sabahi et al., 2018b; Zhou et al., 2018; Zhang et al., 2018b; Dong et al., 2018; Yadav, 2018; Wang et al., 2019a; Zhong et al., 2019; Pilault et al., 2020; Singh et al., 2017; Xu et al., 2019; Magooda et al., 2020; Zhou et al., 2020; Zhong et al., 2020; Wang et al., 2020a). In the general case, extractive summarisation approaches usually show better performance compared to the abstractive approaches, especially when evaluated using *ROUGE* metrics (Kiyoumarsi, 2015). Extractive text summarisation has benefited from the advances in neural networks. Several cutting-edge approaches that leverage the power of neural networks and deep learning have been developed (Filippova et al., 2015; Narayan et al., 2017; Nallapati et al., 2016a; Nallapati et al., 2016b; Verma et al., 2019; Sinha et al., 2018; Wu et al., 2018; Brito et al., 2019; Cheng et al., 2016).

In a nutshell, extractive methods gained popularity for text summarisation due to their simplicity and the high number of available highly extractive summarisation datasets.

2.2.3.2 Abstractive Summarisation

The abstractive summarisation approach involves rewriting the summary (Rush et al., 2015; Liu et al., 2015); and has seen substantial recent gains due to neural sequence-to-sequence models (Chopra et al., 2016; Nallapati et al., 2016a; See et al., 2017; El-Haj et al., 2019b; Paulus et al., 2018). Abstractive summarisation applies language models and natural language generation techniques with the aim of mimicking human summarisation methods. Hence, the algorithm can add words and phrases not present in the original document in order to enhance the meaning of the summary. Abstractive approaches rely on a word-level attention mechanism to determine the most relevant words to the target words at each decoding step. (Wang et al., 2019b; Schüller et al., 2020; Rush et al., 2015; Tan et al., 2017; Nallapati et al., 2016c; Chopra et al., 2016; Tan et al., 2017; Nema et al., 2017; Dohare et al., 2017; Fan et al., 2018; Hasselqvist et al., 2017; Cibils et al., 2018; Chang et al., 2018; Zhang et al., 2018a; Liu et al., 2015; Song et al., 2018; Al-Sabahi et al., 2018a; Lebanoff et al., 2020; Li et al., 2020b; Gunel et al., 2020; Chowdhury et al., 2020b; Saito et al., 2020; Wenbo et al., 2019; Li et al., 2018a; Dohare et al., 2018; Li et al., 2018b; Xie et al., 2018; Liao et al., 2018; Hardy et al., 2018; Li et al., 2018d; Nikolov et al., 2020; Laban et al., 2020; Zhou et al., 2017). Abstractive models can be more concise by generating system summaries from scratch in a context where gold standard summaries were written by human annotators. However, this method suffers from some drawbacks. It can take a long time to converge, and it can be inaccurate in the process of encoding long documents and long sequences, which is exactly the case with UK financial annual reports (above 50,000 tokens per report). Abstractive models also suffer from redundancy, especially when generating summaries of long documents (Cohan et al., 2018), and they may do some hallucination by inventing wrong information that are not mentioned in the original document. In addition training abstractive models face a huge issue which is the lack of well labelled abstractive datasets. That is why this type of summarisation is more challenging and creates difficulties in assessing the quality of the summary.

Like extractive text summarisation, abstractive summarisation has benefited from the advances in neural networks. Shi et al. (2020) gave an exhaustive review of various seq2seq models for abstractive text summarisation from the perspectives of network architectures, training approaches and summary production methods.

Finally, we should mention that we can opt for a hybrid summarisation technique. In fact, several studies such as (Widyassari et al., 2022; Tretyak et al., 2020) proposed to combine extractive and abstractive techniques in order to improve performance.

2.2.3.3 Hybrid Summarisation

Hybrid summarisation refers to a technique where we summarise an input document using multiple techniques, methods, or models. It can combine a mixture of extractive and abstractive methodologies to generate concise text summaries. The idea behind this approach is to combine the strengths of different methods to improve the efficiency of the summarisation model (Bishop et al., 2022; Reda et al., 2022; Muniraj et al., 2023; Elsaid et al., 2023; Xiao, 2023; Narrain et al., 2023; Khatri et al., 2018).

Bishop et al. (2022) proposed a hybrid unsupervised abstractive-extractive summarisation. They go through the document and produce salient text snippets that reflect its main ideas. Subsequently, they will identify the most crucial sentences in the original text by comparing them to the generated texts using BERTScore. This approach allowed them to determine which sentences are most similar and therefore most important within the document.

For example, Reda et al. (2022) presented a novel sequential hybrid model called A3SUT, which combines two main approaches for summarisation. The first approach is extractive, selecting the most important sentences from the original text (using various word embedding techniques and using AraBert as a transformer). Then, the second approach is abstractive summarisation to refine the extracted sentences (they used the mT5 pre-trained transformer model). The two approaches are applied sequentially.

Di Fabbrizio et al. (2014) presented a hybrid method to generate summaries of product and service reviews. By combining natural language generation with sentence selection techniques, Their system takes in written reviews, along with ratings for specific topics, and generates a comprehensive summary that captures the opinions expressed in those reviews. They provide evidence that, compared to extractive methods, summaries generated with hybrid summarisation approaches are more readable and concise.

Elsaid et al. (2023) proposed a combination approach for summarising Arabic texts automatically. The summary is generated through the utilisation of seq-to-seq with attention, bi-directional LSTM, and mT5 transformers.

2.2.4 Financial Summarisation

The employment of summarisation and natural language processing techniques in general has promising applications in the financial domain (El-Haj et al., 2019a). There are many challenges involved in automatically summarising financial reports. For example, reports often contain a lot of technical jargon and numbers. This can make it difficult for automatic summarisation methods to understand the reports. Additionally, financial statements can be long and complex. This makes it challenging for automatic summarisation methods to identify the most essential information in the reports. Despite these challenges, automatic financial narrative summarisation is a promising area of research.

Financial report summarisation is essential for companies and investors for several reasons: It provides a concise and easy-to-understand overview of a company's financial performance and allows investors and analysts to quickly understand the key points in a financial report. It allows companies to provide a concise overview of their financial performance, which is helpful for both shareholders and potential investors. This would help investors to identify trends and make informed decisions about where to invest their money. Finally, financial report summarisation can help to identify trends and potential problems early on, allowing for corrective action to be taken before it is too late.

In conclusion, summarising financial reports is vital for any company or investor looking to stay on top of their finances. It provides crucial information in a concise and easy-to-understand format and can help to identify problems early on.

Gold-standard summaries are essential for objectively assessing an automated system's performance. Evaluation metrics use gold-standard summaries (usually created by humans) to measure the quality of a machine-generated summary. The accessibility of these summaries is contingent on language and field of application. Access to gold standard summaries is very difficult for some tasks, particularly those within a specialised domain such as finance.

What is a gold standard summary for a UK financial annual report? An excellent gold standard summary for a UK financial annual report would include key financial highlights such as revenue, profit/loss, and earnings per share (EPS).

It would also include key performance indicators and information on key business developments and any notable changes in the company's financial position, such as changes in debt or assets. Additionally, it would provide an overview of the company's performance compared to industry benchmarks and any significant risks or uncertainties facing the company. Overall, the summary should provide a clear and concise overview of the company's financial performance, position, and key information for investors to make informed decisions.

2.2.5 Long Document Summarisation

The ability to summarise long documents accurately has become increasingly important and gained momentum within the research community. Whether a professional needs to read long reports or you are a researcher looking to save time on scientific documents, long document summarisation can be a game-changer. In this section, we will explore some of the latest research advancements in the field. Xiao et al. (2019) introduced a neural single-document extractive summarisation model for long documents, which takes account of both the overall and the particular context within each topic. They tested this model on two sets of scientific papers, Pubmed and arXiv, and it surpassed previous extractive and abstractive models when evaluated using ROUGE-1, ROUGE-2 and METEOR scores. As expected, their method also resulted in more advancements when applied to longer documents. Chen et al. (2018) has created a summarisation model which first selects important sentences and then compresses and rephrases them to generate a concise summary. They used an original sentence-level policy gradient method to link two neural networks in a hierarchical manner. Results indicate their model is superior in all metrics (including human assessment) on the CNN/Daily Mail dataset. Cui et al. (2021) introduced the sliding selector network with dynamic memory as a solution to the length limitation of text encoders in neural-based summarisation models. This network utilises a sliding window to segment and extract summary sentences from long documents while also using memory mechanisms to store and update the history information throughout. Results from two large-scale datasets of scientific papers indicate that the proposed model is superior to current state-of-the-art approaches. Moreover, Ying et al. (2021) presented their system submitted to the LongSumm task, which is a shared task on generating long summaries for scientific documents. This is a difficult challenge as traditional language models such as BERT have limited memory and cannot process long inputs. To address this issue, they present SBAS (Session Based Automatic summarisation Model), which uses session and ensemble techniques to generate longer summaries. Their model achieved the highest score in the LongSumm task. In addition Kaushik et al. (2021) took part in LongSumm 2021. This paper (Kaushik et al., 2021) presents their extractive summarisation technique used to solve the challenge. They used the TextRank algorithm and the BM25 score was used as similarity function. Even though, Texrank is a graph-based ranking algorithm that does not need any learning, it still delivers satisfactory results for them with limited computing power and time. Several other papers targeted long document summarisation. Cohan et al. (2018) presented the first neural abstractive summarisation model designed to summarise long documents, such as research papers. Their model features a hierarchical encoder to capture document discourse structure and an attentive decoder to generate a summary. Their results on two large-scale datasets of scientific papers demonstrate that their model surpasses existing models in performance.

2.2.6 Key Challenges for Long NLP Sequences

There are different challenges that come with long documents (the length of the input document is above 512 tokens). The biggest constraint would be to annotate the training dataset by human experts and have a well-labelled training dataset. Furthermore, most of the pretrained open-source language models have a limited number of input tokens (e.g. BERT is limited to 512 tokens). Another challenge is the fact that important information is spread over the whole document which makes it difficult to locate and combine it algorithmically. And finally, hyperparameter tuning while training large language models is very costly. Recent advances in efficient transformers open the door to long-sequence NLP and show promising improvements over shorter baselines while being easier to use. Hence we can adapt and re-use shorter models to bootstrap training longer ones.

2.3 Financial Narratives

2.3.1 What are Financial Narratives ?

Financial narratives represent all financial textual documents produced by firms. Financial narratives could describe the past, current, or projected performance of the company. They could also present innovation plans, research and development strategy and future planned investments of the firm. Financial narratives could be influenced by several factors, such as the country where the company is listed, the language of reporting and the background of the board members. However, these financial narratives are not objective because managers are hesitant to disclose negative information about the company. Their main objective is to boost the stock price and attract the necessary funds to accomplish the firm's strategic objectives.

Since the last few years, there has been an increasing interest in financial narrative processing (FNP), which is a sub-field of Natural Language Processing (NLP) and linguistics focusing on automatic processing of financial narrative data. This interest is shown by the organisation of the financial narrative processing workshop series (El-Haj et al., 2022b; El-Haj et al., 2021a; El-Haj et al., 2020a; El-Haj et al., 2019c; El-Haj et al., 2018). We can distinguish different types of financial narratives (E.g. Periodic information, Ongoing information, Regulatory information, Event-related information).

2.3.2 Different Financial Narratives Structures

French firms regulated by the French Financial Market Authority[2] (Autorité des marchés financiers) report using PDF or HTML files. The European Single Electronic Format XHTML with XBRL tags has been required since 2021.

Regarding the UK markets, all UK firms listed on the London Stock Exchange (LSE) are regulated by the Financial Conduct Authority (FCA) and communicate using free structure PDF format with a variety of different structures (plain text). UK financial narratives present a problem of nomenclature since they do not use a standardised naming convention for different sections making automatic structure detection a real research challenge (El-Haj et al., 2019a; El-Haj et al., 2014).

In Saudi Arabia, firms are regulated by the capital markets authority[3]. The biggest Saudi Arabia firms (Eg. Saudi Aramco, Al Rajhi Bank, SABIC and STC) communicate bilingually in Arabic and English. The Arabic language is mandatory for listed KSA firms. The Arabic reports are designated for the regulator and local investors, while the English version is for foreign investors. KSA's financial narratives have a free structure (published in PDF file format) with no predefined structure.

In the US, firms use a predetermined structure imposed by the regulator (the Securities and Exchange Commission), which requires that firms fill a standard pre–labelled reports template, which they publish in HTML file format. There are different forms used in the US, such as the 10-K and 8-K forms, making US financial narratives very well structured and allowing an easy application of different NLP techniques (El-Haj et al., 2019a).

In the next table, we include a benchmarking explaining some of the differences between four reporting systems: French, UK, KSA and USA markets.

[2] https://www.amf-france.org

[3] https://cma.org.sa/

Financial communications

Comparison :

French market	***UK market***	***KSA market***	***US market***
French regulator: AMF (source of these information) *https://www.amf-france.org/* *French association of investor relation professionals.* *https://cliff.asso.fr/en* ***Format**: HTML or PDF (2020) , Only XHTML with XBRL tags (starting from 2021) → Good news for financial NLP researchers.* Deadline for filling: The annual financial report must be filed, published and disseminated no later than four months after the close of the financial/fiscal year **Different narratives:** Reports communicated annually: -Annual Turnover communication. -Annual net income. -Annual financial report. -Management report: capital and shareholding. -Risk management report -Enterprise Governance report. -CSR report. -Extra financial performance declaration (mandatory for high market caps) -Report on sums to be paid to the government.	UK regulator: financial conduct authority https://www.fca.org.uk **Format**: PDF format with a variety of different structures. Problem of nomenclature: no standardized naming convention exists for different sections in UK annual reports so that even firms adopting the same underlying structure and content may use different terminology to describe the same section(s). [Mahmoud El-Haj* , Paul Rayson] **Public and investor Relations:** firms make a regular release of all their announcements and regulatory filings.	KSA regulator: capital markets authority of Saudi Arabia. https://cma.org.sa The biggest Saudi Arabia firms (Saudi Aramco, Al Rajhi Bank, SABIC, STC) communicate bilingually in Arabic and English. The Arabic language is mandatory for listed KSA firms. The Arabic reports are designated for the regulator and English for foreign investors (especially US). Financial narrative structure: free structure: PDF files: no predefined structure (containing pictures and a lot of colors) → difficult to detect the structure. It is up to the company to define the content of the report **Different narratives:** Annual financial communications -Full year financial report. -Full year results press release. -Full year results webcast presentation. -Yearly ESG reports.	*US regulator: US securities and exchange commission.* *https://www.sec.gov* US Financial communication society https://thefcs.org list of all available forms: https://www.sec.gov/forms **Format**: plaint text: US Securities and Exchange Commission forces firms to fill documents and follow a standard format and a pre–labeled annual reports template which they publish in HTML file format. → US financial narratives are very structured and allow an easy application of NLP techniques. **Different narratives:** -The 10-K Form (annual report): Quoted companies are requested to annually publish financial reports in plain text format. The structure with a standard set of headings, is rigid and imposed by the SEC. This allows a standard structure to be followed by each company making it easy to extract information and easily detect structure

Bi annual communications: -Bi annual financial reports Quarter communications: -Quarterly (or interim) financial reports. Monthly communications: -Issuers publish every month and transmit to the AMF the total number of shares and voting rights making up their share capital, if this number has varied compared to that previously published -News release or consultation of information for the general meeting. Other communications (Fr or EN): -Pro forma information (report of the company's earnings that excludes unusual or nonrecurring transactions. -Estimated financial data. -Bond issue. -Profit warning. **Channels to communicate**: AMF website, firm website, Press agencies, data providers, Social media →High level of regulation →Information should be available for the last 10 years for public.			-Non-IFRS measures reconciliation and definitions. -Risk management report -Sustainable Development reports Quarter financial communication -Intermediary financial report. -Company corporate review. Other communications: IPO documents: -Announcement of offer price. -Announcement of offering size and final offer price Prospectus.	-The 8-K Form of Security and Exchange Commission: (SEC) requires the publicly-traded company to file Form 8-K (also called 'Material Event Report' or 'Current Report') when certain types of the corporate event take place. In general, an 8-K report should be filed when the company has an event that its shareholders should be aware of. -Pre-earning announcement . -Earning announcement. -Conference calls -CSR (Corporate social responsibility) reports -Audit reports. -IPO (initial public offer) prospectus -ICO (initial Coin Offer) prospectus

Classical Channels for Financial communications:	*Alternative channels for financial communications:*
• Regulators. • Firm website. • Financial news agencies. • Press. • Financial rating agencies. • real-time information systems (Thomson Reuters, Bloomberg).	• Blogs. • Microblogs (Twitter). • Social and professional networks (Facebook, LinkedIn, Google+) • Forums. • Image and video-sharing platforms (YouTube, Instagram, Vine, Pinterest). • Magazines (Investor relation Magazine)https://www.irmagazine.com

2.3.3 Different Sources of Narratives

Financial narratives are communicated through different communication channels. The company's website is still the key source of information for retrieving all documents related to a company. In addition, there are other channels such as the websites of financial regulators, financial news agencies, financial Press and real-time information systems (e.g. Thomson Reuters Refinitiv, Bloomberg, EMIS and Factivia). In other countries, there are centralised financial data lakes where we can find all the narratives related to listed companies in this specific market. Furthermore, there are other alternative means of communication such as Financial news magazines (FT, Fortune, Forbes, Financial Post), Financial blogs, and social media (especially X, LinkedIn), where firms are more involved.

2.4 Related Work and Literature Review for Arabic Language

2.4.1 Arabic Natural Language Processing

Arabic Natural Language Processing (ANLP) is a field of study that focuses on developing computational models and techniques to analyse, understand, process and generate human language in Arabic. ANLP involves various tasks such as text classification, sentiment analysis, part-of-speech tagging, named entity recognition, machine translation, morphological analysis and tokenisation, speech recognition and text-to-speech, question answering, text generation and text summarisation, among others. Arabic, being a complex language with rich syntax, morphology, and semantics, presents unique challenges and opportunities for NLP research. However, significant progress has been made in recent years, with the development of Arabic-specific NLP conferences and workshops, Arabic-specific NLP tools, resources, and datasets. Researchers have explored different approaches, including statistical models, and deep learning techniques, to overcome these challenges and improve the performance of Arabic NLP systems. The primary aim of these applications is to tackle the complex issues related to the structure and meaning of the Arabic language (Guellil et al., 2021; Shaalan, 2010; Farghaly et al., 2009; Zitouni, 2011; Abo et al., 2019; Ghallab et al., 2020; Dhouib et al., 2022; Shaalan et al., 2018).

2.4.2 News Summarisation

Filippova et al. (2009) proposed a system for summarising financial news which takes as its input the name of a company. The news related to that company is retrieved from Yahoo News and then sentences are ranked in terms of relevance and significance. Additionally, Berger et al. (2000) made use of statistical techniques to create query-based summaries by using frequently asked questions documents on websites; each answer in the FAQs is taken as a summary relative to the question that preceded it. Lopyrev (2015) used an encoder-decoder recurrent neural network with long short-term memory (LSTM) units and attention to generate headlines from the text of news articles. This model demonstrated the ability to accurately condense news stories into concise headlines. Additionally, the function of different neurons in a simple attention mechanism was examined to observe how the neural network chooses which input words to focus on. Surprisingly, this basic attention mechanism had better performance than a more intricate one on a test set of articles.

2.4.3 Arabic Pretrained Language Models

Pre-trained language models have gained popularity in Natural Language Processing (NLP). Despite their success, most are trained on English or multilingual datasets. Leveraging the vast amount of unlabelled data available online, language models provide an efficient way to pre-train continuous word representations that can be fine-tuned for a downstream task, along with their contextualisation at the sentence level. Generally, pre-trained models are trained on massive corpora using graphics processing units (GPUs) or the more advanced tensor processing units (TPUs). Most follow the transformer architecture proposed by Vaswani et al. (2017). Recently, a number of Arabic language models have been developed. In this subsection, we will present some of the common Arabic pretrained language models:
AraBERT (Antoun et al., 2020) was trained using the same architecture as BERT (Devlin et al., 2019) and used the BERT Base configuration. AraBERT was trained on 23GB of Arabic text, making approximately 70M sentences and 3B words from Arabic Wikipedia, the Open Source International dataset (OSIAN) (Zeroual et al., 2019), and (El-Khair, 2016) Corpus (1.5B words). Antoun et al. (2020) compared the performance of AraBERT to multilingual BERT from Google and other state-of-the-art models. Their results prove that AraBERT performs state of the art on most tested Arabic NLP tasks.
ARBERT (Abdul-Mageed et al., 2021) is a large-scale pre-trained masked language model for modern standard Arabic. To train ARBERT, Abdul-Mageed et al. (2021) used the same architecture as $BERT\_base$: 12 attention layers. It has approximately 163M parameters and was trained on a 61GB collection of Arabic datasets.
MARBERT (Abdul-Mageed et al., 2021) is a large-scale pre-training masked language model for Dialectal Arabic (DA) and MSA. MARBERT was trained on 1B Arabic tweets.
AraBART (Eddine et al., 2022) is an Arabic sequence-to-sequence model where the encoder and the decoder are trained end-to-end. It is based on BART. AraBART follows the architecture of $BART\_Base$, which has 6 encoder and 6 decoder layers and 768 hidden dimensions. AraBART has 139M parameters and achieved state-of-the-art results on multiple abstractive summarisation datasets.
AraT5: (Nagoudi et al., 2022a) is the first Arabic text to text model. It is similar to T5 (Raffel et al., 2020) in terms of structure and size. Nagoudi et al. (2022a) released three powerful Arabic text-to-text Transformer versions. The model was trained on more than 248GB of data, including 70GB MSA and 178GB tweets. For

evaluation, they used an existing benchmark for Arabic language understanding and introduced a new benchmark for Arabic language generation (ARGEN).

Multi-dialect-Arabic-BERT (Talafha et al., 2020) is a model developed by Mawdoo3 AI team. It was trained for the Nuanced Arabic Dialect Identification (NADI) shared task. It represents an ensemble of different training iterations of a pre-trained BERT model.

JABER and SABER: Junior and Senior Arabic BERT (Ghaddar et al., 2021) found that most of the released Arabic BERT models were under-trained. So they presented JABER and SABER, Junior and Senior Arabic BERT models. Experimental results show that their models achieve state-of-the-art performances on ALUE, a new benchmark for Arabic Language Understanding Evaluation.

AraELECTRA (Antoun et al., 2021) is a language model for Arabic that uses an ELECTRA objective (Clark et al., 2020).

ArabicBERT: Safaya et al. (2020) developed an Arabic BERT-based model called ArabicBERT, which consists of four variants: ArabicBERTMini, ArabicBERTMedium, ArabicBERTBase and ArabicBERTLarge. These models were trained on unshuffled data from OSCAR (Suárez et al., 2019), Wikipedia in the Arabic language and other sources of text totaling 95GB (approximately 8.2B tokens).

Arabic-ALBERT (Safaya, 2020) is a language representation model based on a lite Bert (ALBERT), which was proposed by Lan et al. (2020). This Transformer-based neural network architecture is analogous to BERT and XLM-R, featuring two techniques to reduce parameters in order to quicken training speed and diminish memory usage of the BERT model. The pretraining for Arabic-ALBERT comes with three architectures available: Arabic-ALBERTBase, Arabic ALBERTLarge, and Arabic-ALBERTXLarge.

CamelBERT (Inoue et al., 2021) was trained with the BERTBase architecture on four distinct Arabic datasets, namely MSA (107GB), dialectal Arabic (54GB), classical Arabic (6GB) and a combination of these three sets (167GB). The vocabulary used for training comprised 30K WordPieces tokens.

QARIB: Chowdhury et al. (2020a) developed the QCRI Arabic and dialectal BERT (QARiB) model. To train it, Chowdhury et al. (2020a) used 97GB of Arabic text with 14 billion tokens, including 180 million sentences in Modern Standard Arabic and 420 million tweets collected via Twitter API that were identified as Arabic. The MSA data for QARIB was a mixture of Arabic Gigaword (Parker et al., 2009) and Abulkhair Arabic Corpus (Abu El-khair, 2016).

Jasmine: Arabic GPT Models for Few-Shot Learning: Nagoudi et al. (2022b) introduced JASMINE, a suite of powerful auto-regressive Transformer language models with parameters ranging from 300 million to 13 billion. They pretrain these models using 400GB of data from multiple Arabic dialects and domains. The performance was assessed on both intrinsic and extrinsic tasks in a comprehensive benchmark for zero-shot and few-shot learning in NLP.

2.4.4 Financial Pretrained Language Models

Finbert (Yang et al., 2020a): is the first BERT model pre-trained on financial communication text. It is trained on a 4.9B tokens corpus composed of corporate reports 10-K & 10-Q (2.5B tokens), earnings call transcripts (1.3B tokens), and analyst reports(1.1B tokens). Finbert is then finetuned for three use cases: sentiment classification task, ESG classification task and forward-looking statement (FLS). This model achieves superior performance on financial sentiment classification tasks.
Finbert[4] (Araci, 2019) is another financial BERT. This model is motivated by the fact that general purpose models are not effective enough on financial text due to the specialised terminology used in a financial context. The results reported by the authors show improvement in all metrics for two financial sentiment analysis datasets.

2.4.5 Arabic Summarisation Datasets

Arabic Gigaword (Parker et al., 2009) is a comprehensive set of Arabic newswires acquired over several years. It could be used for a newswire headline-generation task.
XLSum (Hasan et al., 2021) is a multilingual text summarisation dataset for 44 languages, including Arabic.
WikiLingua (Ladhak et al., 2020) is a new benchmark dataset for Cross-Lingual abstractive summarisation. It includes 18 languages, including Arabic.
Kalimat (El-Haj et al., 2013) is an Arabic natural language resource that consists of 20,291 Arabic articles collected from an Omani newspaper. It is composed of extractive single-document and multi-document system summaries.
EASC (Essex Arabic Summaries Corpus (EASC) Dataset) contains 153 Arabic articles and 765 human-generated extractive summaries of those articles.

2.4.6 Arabic Summarisation Work

Previous Arabic summarisation work focused on extractive approach (EL-Haj et al., 2011; Douzidia et al., 2004; Tanfouri et al., 2021; Lagrini et al., 2021; Haboush et al., 2012; Ayed et al., 2021; Belkebir et al., 2015). New research papers applied multilingual models to Arabic abstractive summarisation. For example, Kahla et al. (2021) trained different neural abstractive summarisation systems for Arabic by fine-tuning multilingual BERT and multilingual BART on a dataset they created themselves. Evaluation of the models was performed using the ROUGE metric with a manual evaluation of adequacy and fluency.

2.4.7 Financial Arabic corpora

BORSAH (Mohammed et al., 2018): An Arabic sentiment financial tweets corpus crawled from Twitter. The dataset consists of (41,455) Arabic gold-standard

[4] `https://github.com/ProsusAI/finBERT`

annotated Twitter feeds gathered from (118,283) tweets tagged manually. The corpus was labelled for sentiment analysis. Three different machine-learning algorithms were applied to parts of the corpus. The authors claim that this is the largest manually annotated Arabic financial tweets corpus for SSA.

ABMC: Arabic in Business and Management Corpora (ABMC) 2016[5]. It comprises 1,200 Arabic articles as plain text and is tagged using Stanford Arabic Part of Speech Tagger.

2.5 Related Work and Literature Review for French Language

2.5.1 French and Multilingual Pretrained Language Models

Recently, research has produced French pretrained general language models. In addition, multilingual pretrained models have also emerged after the release of popular English pretrained models. In this section, we describe the French language models and multilingual models that we adopted for our monitoring study.

For French pretrained models, we used CamemBERT[6] (Martin et al., 2020), a French version of BERT. Martin et al. (2020) presented one of the highest performing French models called CamemBERT. Martin et al. (2020)'s paper assesses the potential of training Transformer-based language models for languages other than English, using French as an example. They evaluate the CamemBERT model's performance on part-of-speech tagging, dependency parsing, named entity recognition and natural language inference tasks. Additionally, they compare the use of web-crawled data with Wikipedia data when it comes to training monolingual models. CamemBERT surpasses state-of-the-art in all four downstream tasks.

Furthermore, we used BARThez[7], the French equivalent of the BART model, which was trained in an encoder-decoder manner. Also we explored mBarthez[8], an extended trained version of mBart[9] Based on the original paper, Barthez is well-suited for generative tasks due to its sequence-to-sequence architecture. For decoder only models we used French GPT (small[10] and base[11]), which is a GPT-2 model pretrained on a very large and heterogeneous French corpus (more than 60Gb). In terms of other French decoder models, we also employed BelGPT-2 [12] in our experiments. As an alternative to monolingual models, multilingual models pretrained on datasets spanning multiple languages, such as mc4 can be used. One of the main advantages

[5] https://sourceforge.net/projects/arabic-business-copora/
[6] https://huggingface.co/camembert-base
[7] https://huggingface.co/moussaKam/barthez
[8] https://huggingface.co/moussaKam/mbarthez
[9] https://huggingface.co/facebook/mbart-large-cc25
[10] https://huggingface.co/asi/gpt-fr-cased-small
[11] https://huggingface.co/asi/gpt-fr-cased-base
[12] https://huggingface.co/antoinelouis/belgpt2

of using a pretrained multilingual model is that it can understand multiple languages, which can save a lot of time and resources. In contrast, training a separate model for each language would be very time-consuming and resource-intensive. Examples of this type of model include mBART, XLM-R, mBERT and mT5. In this chapter, we will focus on utilising both mT5 and the recently released mBARThez. This latter model was developed by the same team behind barthez after further training of mBART using an MLM training goal which made it suitable for French generative and discriminative tasks. Table 2.1 presents more details on the set of models that we used in this study. We give the model description, number of parameters, architecture and the pre-training corpus.

Other French Pretrained Language Models non-used in this Study:
Flaubert: Le et al. (2020) introduced FlauBERT, which is a model learned from an extensive and diversified French corpus and trained using the Jean Zay supercomputer from the CNRS (French National Centre for Scientific Research). They trained models of varying sizes. The trained models were evaluated on various NLP tasks (text categorisation, paraphrasing, natural language inference, parsing, word sense disambiguation). In most cases, they achieved better results than with alternative pre-training techniques.
Pagnol (An Extra-Large French Generative Model): Launay et al. (2022) introduced PAGnol, a family of models, which is a collection of four French GPT-like models. They have trained four models on CCNet, the largest one having 1.5 billion parameters (the largest model ever created for the French language), and released them publicly. Launay et al. (2022) claim it is the most expansive non-sparse French language model that has been trained without pre-existing data.
Boris: Müller et al. (2022) introduced a larger French auto-regressive language model, Boris which is trained on French data starting from the English model GPT-J (Wang et al., 2021) while PAGnol was trained from scratch. Boris is a 6B parameter autoregressive language model based on the GPT-J architecture and trained using the mesh-transformer-jax. Boris was trained on around 78B tokens of French text from the C4 dataset. Müller et al. (2022) findings demonstrate that their model surpasses existing French language models and is comparable to GPT-3 on a variety of French zero-shot benchmarks. Simoulin et al. (2021) presented a Generative pretrained Transformer in French. It is a French version of the GPT model. GPT models are pretrained with self-supervised objectives and consequently designed for a particular language. The first corpus, used for training GPT-fr-124M, is an aggregation of existing corpora: Wikipedia, OpenSubtitle and Gutenberg. The documents are separated into sentences.
JuriBERT: Douka et al. (2021) investigated the potential of adapting a BERT model to French legal texts with the aim of aiding law professionals. Results revealed that certain tasks do not gain much from general language models that are pretrained on large quantities of data. They explored the use of smaller architectures in specialised sub-languages and their advantages for French legal text. Evidence suggests that domain-specific pretrained models may be more efficient than generalised ones when applied to the legal domain.

Model	Description	#params / architecture	Pre-training corpus
BARThez Base	encoder-decoder French pretrained model	165M, architecture: BASE, layers: 12	French part of CommonCrawl and NewsCrawl
mBARThez	continuation of the pre-training of multilingual BART	450M, architecture: large, layers: 24	BARThez' corpus
BelGPT2	a GPT-2 model pre-trained on a large French corpus	Small	CommonCrawl, NewsCrawl, Wikipedia, Wikisource, Gutenberg...
CamemBERT	a Tasty French Language Model based on the RoBERTa model.	110M / 335M for Base/Large versions	OSCAR: a multilingual corpus obtained by language classification and filtering of the Common Crawl corpus.
GPT FR Cased[a]	Decoder only French Pretrained transformer suitable for generative tasks	124 M / 1,017 B for small / Base versions	A very large heterogeneous French corpus composed of: Wikipedia, OpenSubtitle, Gutenberg and Common Crawl.
mT5	a multilingual version of T5	300M – 13B	Common Crawl-based dataset mC4 (101 languages)
mBART	multilingual BART model with 12 encoder and decoder layers	600M	+25 languages monolingual corpus (CC25)
ByT5[b]	ByT5 is a tokeniser-free version of Google's T5 and generally follows the architecture of mT5.	small and Base versions	ByT5 was only pretrained on mC4, excluding any supervised training with an average span-mask of 20 UTF-8 characters.
Xprophetnet[c]	ProphetNet is a new pretrained language model for sequence-to-sequence learning with a novel self-supervised objective called future n-gram prediction.	Large	pretrained on wiki100 xGLUE dataset.

Table 2.1: Overview French Language Models and Multilingual Models.

[a] `https://huggingface.co/asi/gpt-fr-cased-base`
[b] `https://huggingface.co/google/byt5-small`
[c] `https://huggingface.co/microsoft/xprophetnet-large-wiki100-cased`

2.5.2 Previous French Summarisation Work

Kamal Eddine et al. (2021) introduced BARThez, a pretrained seq2seq model for French that is based on BART and especially well-suited for generative tasks. Kamal Eddine et al. (2021) assess BARThez on five FLUE benchmark[13] discriminative tasks and two generative tasks from their newly created OrangeSum summarisation dataset. Results show that BARThez is highly competitive with state-of-the-art BERT-based French language models such as CamemBERT and FlauBERT. In addition, the paper (Kamal Eddine et al., 2021) presents mBarthez by continuing training of the multilingual BART model on BARThez's corpus and found that mBARThez model significantly enhanced the output quality of BARThez.
Zhou et al. (2022) explores abstractive summarisation of oral dialogues in French using several language-specific pretrained models such as BARThez and BelGPT-2 as well as multilingual pretrained models like mBART, mBARThez and mT5. Testing was done on the DECODA (Call Center) dialogue corpus, which is used to create summary descriptions of conversations between a caller and one or multiple agents, depending on the context. The results demonstrate that BARThez models are superior to prior comparable methods in terms of performance on DECODA. Additionally, they evaluate the limitations of pretrained models and identify obstacles that need to be addressed when summarising spontaneous dialogues(meetings, interviews or phone calls).

2.6 From Neural Networks to Transformers

2.6.1 Sequence to Sequence Models

Sequence-to-sequence (seq2seq) models constitute a common framework for solving sequential problems (Sutskever et al., 2014b; Keneshloo et al., 2019). Sequence-to-sequence models are used in a variety of tasks such as text summarisation, machine translation, headline generation and speech-to-text. The seq2seq framework is mainly achieved using neural networks where both input and output are sequences of varying lengths.

2.6.2 Recurrent Models

When it comes to Recurrent Neural Networks (RNNs), they have a natural ability to consider the order of words. They dissect a sentence one word at a time, sequentially.

The encoder hidden state math at time step t, with input token embedding:

$$h_t = RNN_{enc}(x_t, h_{t-1}) \tag{2.1}$$

[13] http://fluebenchmark.com/

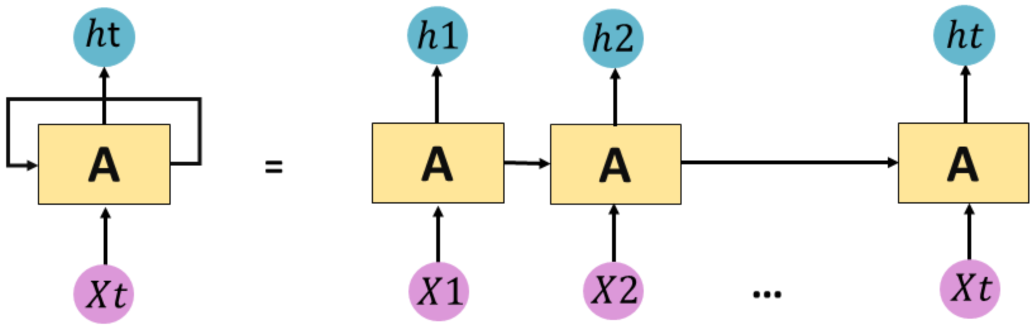


Figure 2.2: A vanilla RNN architecture. reproduced from: (Christopher, 2015).

The decoder hidden state math at time step t, with input token embedding:

$$s_t = RNN_{dec}(y_t, s_{t-1}) \tag{2.2}$$

2.6.3 Encoder-Decoder

The Encoder-Decoder model is a method for predicting sequences, where the probability of a certain output given an input is estimated using two neural networks: an encoder and a decoder. These two components are jointly trained with the aim of minimising a loss function (or a reward function), such as the log-likelihood or the loss entropy based on N pairs of input and output sequences (Cho et al., 2014). The definition of the encoder-decoder framework with respect to temporal data was presented by Bahdanau et al. (2016) (see Eq. 2.3). The encoder-decoder was initially implemented using RNN networks (Cho et al., 2014; Srivastava et al., 2014). Later, it was implemented using convolutional layer networks (Gehring et al., 2017), and finally, it is now mostly implemented through the attention mechanism (Vaswani et al., 2017).

$$\zeta = -\frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{T} log(p(y_i^k|y_0^k, ..., y_{i-1}^k, x^k)) \tag{2.3}$$

2.6.4 Attention Mechanism

The **Attention Mechanism** has achieved great success and is commonly used in seq2seq models for different natural language processing (NLP) tasks, such as machine translation and neural text summarisation. In fact, Attention is a layer of calculations that lets your model focus on the most essential parts of the sequence for each step. Queries, values, and keys are representations of the encoder and decoder hidden states, and they are used to retrieve information inside the attention layer by calculating the similarity between the decoder queries and the encoder key-value pairs. So, the *attention mechanism* uses encoded representations of both the input or the encoder hidden states and the outputs or the decoder hidden states. The **keys** and **values** are pairs of dimension N, where N is the input sequence length and comes from the encoder's hidden states. Figure 2.3 shows an overview of the attention mechanism. The Attention mechanism is self-sufficient for dealing with text data and does not necessitate any Recurrent Neural Network (RNN) in its design.

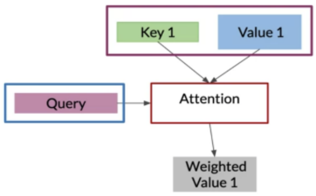


Figure 2.3: Attention mechanism. (source (Mourri et al., 2021))

We can differentiate three types of attention:

- *Encoder-decoder* attention when a sequence of tokens can connect to another one.
- *Bi-directional* self-attention where words in the same sentence can look at previous and future words.
- *Causal attention* where in the same sentence, words attend to words in the past (queries and keys come from the same sentence). In fact, causal attention does not allow words to attend to words in the future since these have not been generated yet. Causal attention could be used for generating text, such as abstractive summaries of long articles.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.4}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2.5}$$

$$e_{ij} = a(s_{i-1}, h_j)j \tag{2.6}$$

Equation 2.4 describes the computation of a context vector c_i for the ith time step of the output sequence. The context vector is calculated as a weighted sum of the encoder hidden states h_j using attention weights α_{ij}

Equation 2.5 calculates the attention weights α_{ij} for each time step j of the input sequence. The attention weight is computed as the exponential of an alignment score e_{ij} , normalised by the sum of exponential alignment scores across all time steps.

Equation 2.6 defines the alignment score e_{ij} between the ith decoder state s_{i-1} (often the previous decoder hidden state or the generated token) and the *jth* encoder hidden state h_j. The function a computes the alignment score based on these two states.

2.6.5 Transformers

The Transformer architecture introduced by Vaswani et al. (2017) was the next step in the encoder-decoder and seq2seq modelling revolution. The Transformer[14] model differs from typical encoder-decoder architectures by employing multi-headed self-attention in place of recurrent layers. The Transformer architecture was introduced as a novel pure attention-only sequence-to-sequence architecture. They generally feature several attention mechanisms with multiple heads, residual connections, layer normalisation, feedforward connections, and positional embeddings. In fact, Recurrent Neural Networks (RNNs) handle the input sequence token by token, which make parallelisation difficult. In the Transformer model, attention mechanisms are utilised instead of using recurrent connections like RNNs and CNNs, which greatly increase the potential for parallel processing. Moreover, transformers do not suffer from vanishing gradient issues connected to sequence length as RNNs do, and finally, the Transformer Encoder Decoder (TED) architecture has dominated the generative AI field since 2017. It allows models to be trained on a large corpus using a high number of parameters, creating a new state-of-the-art model in text summarisation, text generation and machine translation. (see Figure 2.4)

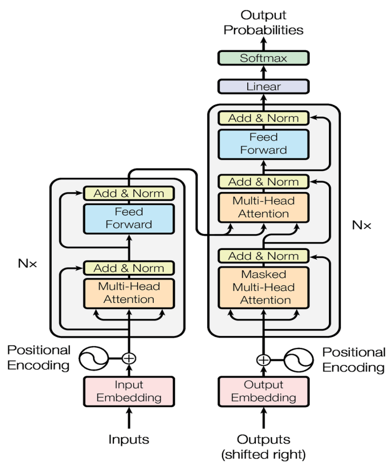
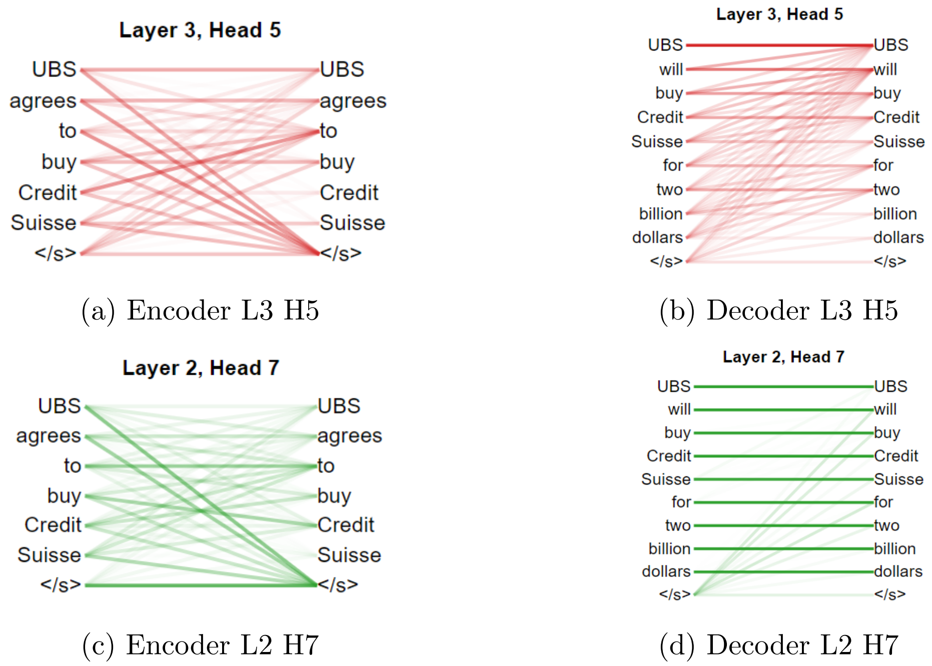


Figure 2.4: Transformer Model Architecture. Source (Vaswani et al., 2017)

Self-attention: Self-attention will mean that every word (or token) in an input sentence will have access (or see) to other words in the same sequence with certain weights for every word. It means every word on the left side sees words on the right side. The weight is high when the colour is dark, and it is low when the colour is

[14]Transformer's library: `https://github.com/huggingface/transformers`

light. Higher weights mean high similarity between the two tokens in the sentence context. (see Figure 2.5)



(a) Encoder L3 H5

(b) Decoder L3 H5

(c) Encoder L2 H7

(d) Decoder L2 H7

Figure 2.5: An example of the attention mechanism. We give two example sentences. We report two different heads and layers from the Encoder and Decoder.

Scaled Dot-Product Attention: The core of the Transformer is based on self-attention. Attention weights come from a dot product scaled function, and a softmax function is needed to convert these weights into probabilities. The Eqn. 2.7 shows the equation of Scaled Dot-Product attention: self-attention involves a query Q and a range of key-value pairs K,V, with an output consisting of a weighted sum of values (Vaswani et al., 2017; Lin et al., 2017). (see Figure 2.6a). To conclude, given queries Q (of dimension d_k), keys K (of dimension d_k), and values V, the scaled dot-product attention computes the attention scores and weighted sum of values as follows in Eqn. 2.7:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.7}$$

Where $\sqrt{d_k}$ is the dimension of the key vector math and query vector math.

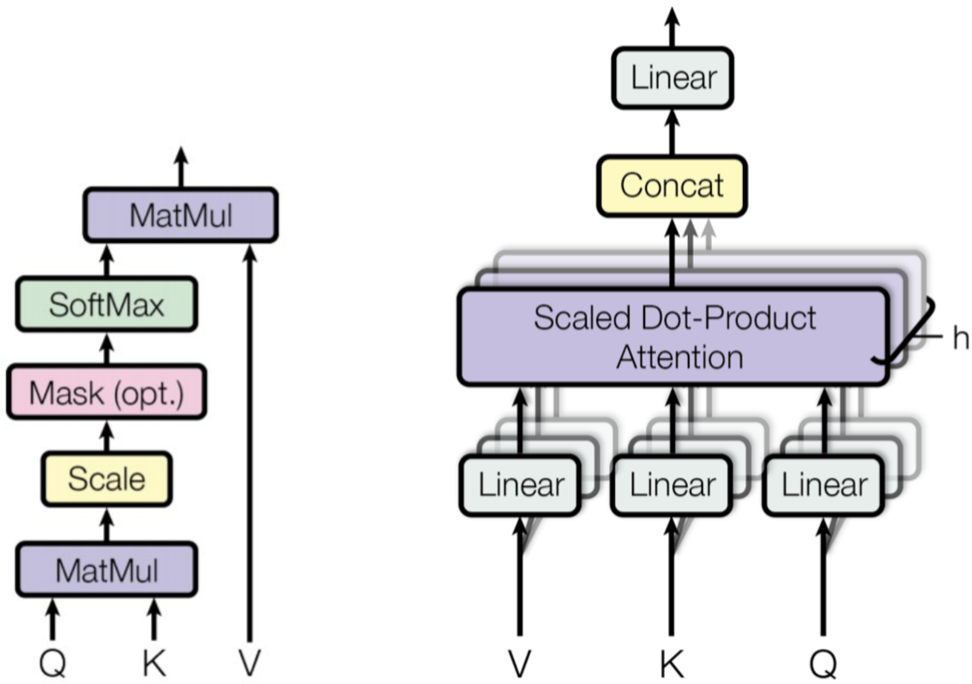
Multi-Head Attention: The Multi-head Attention module is designed to use attention mechanisms several times simultaneously in a parallel execution strategy. So, Multi-head attention involves multiple parallel attention operations, followed by concatenation and linear transformation (Vaswani et al., 2017). The Eqn. 2.8 shows the equation of multi-head attention. Given a word, we take its embedding, then we multiply it by the Q, K, V matrix to get the corresponding queries, keys and values. When we use multi-head attention design, each of the attention-pooling outputs is a head, and the final result is the concatenation of different heads (Vaswani et al., 2017). In a nutshell, multi-head attention linearly projects the queries, keys and values h times. We perform the attention function in parallel on each of these projected versions of queries, keys and values. Then, these are concatenated and projected, resulting in the final values (see Figure 2.6b)

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2.8}$$

Where each *head_i* is computed using the scaled dot-product attention.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2.9}$$

where $\mathbf{Q} \in \mathbb{R}^{\mathbf{d_q}}$, $\mathbf{K} \in \mathbb{R}^{\mathbf{d_k}}$, $\mathbf{V} \in \mathbb{R}^{\mathbf{d_v}}$
and $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$



(a) Scaled Dot-Product Attention. Source (Vaswani et al., 2017)

(b) Multi-Head Attention. Source (Vaswani et al., 2017)

Figure 2.6: (Scaled Dot-Product Attention (Left). Multi-Head Attention (right). Source (Vaswani et al., 2017)

Positional Encodings Positional Encoding is a fundamental mechanism when dealing with transformer models. Positional Encoding is added to the input embeddings at the bottoms of the encoder and decoder stacks. It has to have the same dimension d_{model} as the embeddings. The purpose behind this is to provide information about the positions of each word or token within a sequence. We should mention that *Positional Encoding* is not an integral component of the model's architecture. In fact, it can be presented as a crucial part of the pre-processing stage. Positional encoding is an essential technique used by the transformer to understand the context of words in a sentence. It helps to indicate the relative positions of words in a sentence to better understand their meaning and relationship to other words. Without positional encoding, it would be difficult to accurately interpret the meaning of a sentence (especially in difficult languages), as the context and relationship between words may be non-trivial (Chen et al., 2021; Kazemnejad et al., 2023; Wang et al., 2020b; Rosendahl et al., 2019).

Eq. 2.10 and Eq. 2.11 show example on how to calculate the positional encoding through sinus and cosines functions of varying frequencies and phases. These functions generate distinct embeddings for each word position. When we combine these words with word embeddings, they provide both content-based and position-based knowledge to the model.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2.10}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2.11}$$

Where:
pos is the position of the token.
i is the dimension of the positional encoding vector.
d_{model} is the dimension of the model's embeddings.

There have been several research projects targeting positional encoding. Chen et al. (2021) suggested new methods to enhance positional encodings, with relative position encodings proving to be more effective. Their analysis demonstrates that the improvement is actually due to transferring positional information from the input to the attention layer. Hence Chen et al. (2021) presented Decoupled Positional Attention for Transformers (DIET), a simple, powerful mechanism for incorporating position and segment information into Transformer models.

Transformer Encoder Layer:

$$\text{EncoderStack}(\mathbf{X}) = \text{MultiHeadAttention}(\mathbf{X}) + \text{FeedForward}(\mathbf{X}) \tag{2.12}$$

Where: $\mathbf{X}$ represents the input sequence. $\text{MultiHeadAttention}(\mathbf{X})$ is the result of applying multi-head self-attention to the input sequence. $\text{FeedForward}(\mathbf{X})$ is the result of applying a position-wise feedforward neural network to the input sequence.

Output of Self-Attention:

$$\text{SA}_1 = \text{MultiHead}(X, X, X) \tag{2.13}$$

Position-wise Feedforward:

$$\text{FFN}_1 = \text{ReLU}(\text{Conv1d}(\text{SA}_1)W_1 + b_1)W_2 + b_2 \tag{2.14}$$

Transformer Decoder Layer: The decoder stack of the transformer is exactly the same as the encoder stack. But it adds an extra multi-head cross-attention layer to link the encoder to the decoder. The decoder works in an auto-regressive manner, using the tokens it has already produced to create the next ones.

$$\mathrm{MSA}_1 = \mathrm{MultiHead}(\mathrm{Y}, \mathrm{Y}, \mathrm{Y}) \tag{2.15}$$

$$\mathrm{EDA} = \mathrm{MultiHead}(\mathrm{MSA}_1, \mathrm{X}, \mathrm{X}) \tag{2.16}$$

$$\mathrm{FFN}_2 = \mathrm{ReLU}(\mathrm{Conv1d}(\mathrm{EDA})W_3 + b_3)W_4 + b_4 \tag{2.17}$$

Transformer Architecture: Following the description of Vaswani et al. (2017), the encoder of the transformer is composed of six identical layers. Each layer has two sub-layers. The decoder is also composed of six identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack (Vaswani et al., 2017). The process of autoregressive generation could be modelled as follows: the encoder will receive an input sequence (x_1, x_2, ..., x_n) and will map it to a vector (x^1, x^2, ..., x^n). After that, the decoder will autoregressively generate an output vector (y_1, y_2, ..., y_n).

2.6.6 Language Models

Pretrained language models are extensively utilised in the field of Natural Language Processing (NLP). Since the release of the first language model architecture in 2017 (Vaswani et al., 2017), there has been a surge in the number and size of released language models. These models have expanded in both size and capabilities due to the utilisation of efficient and adaptable Transformer architecture instead of recurrent neural networks, as introduced by Vaswani et al. (2017). By means of self-supervised learning, these models acquire contextual word embeddings instead of static embeddings (word2vec, GloVe), enabling them to develop a comprehensive comprehension of language. Following the completion of pretraining, they can be further finetuned and tailored for specialised tasks such as classification or summarisation. One of the main axes of difference between transformers is the training objective (e.g., masked language modelling and seq2seq objectives suited to conditioned generation tasks). Also, transformers can be pretrained as encoder-only (e.g., BERT), encoder-decoder (e.g., T5), and decoder-only (e.g., GPT series) (see Figure 2.7):

- Bidirectional Encoder-Only models such as Bert (Devlin et al., 2019) (trained on masked language modelling) useful for discriminative tasks. A Transformer encoder consists of self-attention layers.

- Encoder-Decoder models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2020).
- Causal autoregressive Decoder-Only such as GPT (generative pre-training) (Radford et al., 2018; Brown et al., 2020). The 1.5-billion-parameter GPT-2 obtained the state-of-the-art results on language modeling benchmarks

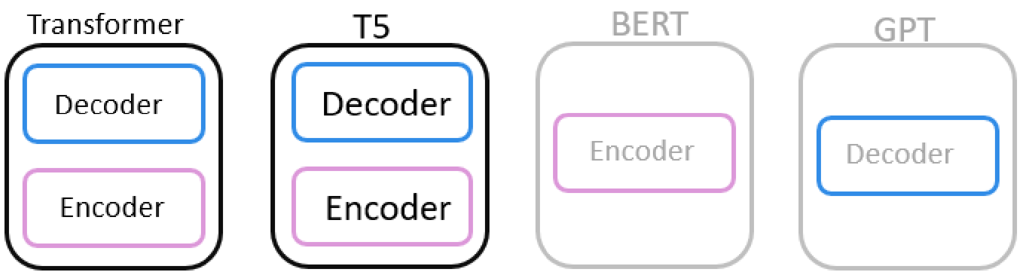


Figure 2.7: Different transformer architectures

Several pretrained language models (PLMs) have been released since 2017. (Chung et al., 2022; Zaheer et al., 2021; Brown et al., 2020; Beltagy et al., 2020; Lan et al., 2020; Sanh et al., 2020; Lewis et al., 2019; Jiao et al., 2020; Dai et al., 2019; Kitaev et al., 2020; Conneau et al., 2020) have become a core component of the natural language processing (NLP) pipeline.

Model's size: language models are getting larger quickly, growing from multi-million to multi-billion parameters (Henighan et al., 2020; Kaplan et al., 2020). Scaling laws demonstrate a clear correlation between the size of the model and its performance, which validates the idea behind the training of more extensive large models (Brown et al., 2020).

Large language models: Smith et al. (2022) used DeepSpeed and Megatron to train Megatron-Turing NLG 530-billion-parameter, one of the world's largest and most powerful generative language models. In addition, Rae et al. (2022) presented *Gopher*, the 280-billion-parameter language model pretrained with 300 billion tokens. After that, Hoffmann et al. (2022) from Deepmind pretrained Chinchilla which is a 70 billion parameters model that outperformed Gopher on many tasks. Another milestone for large language models was PaLM (a 540-billion-parameter transformer decoder) and PaLM 2 (Chowdhery et al., 2022; Anil et al., 2023). We can see that the number of parameters is increasing exponentially.

At the time of writing, we have a panoply of large language models such as OPT: Open Pre-trained Transformer Language Models (Zhang et al., 2022), BLOOM: a 176B-Parameter Open-Access Multilingual Language Model (Workshop et al., 2023), LLaMA: Open and Efficient Foundation Language Models (Touvron et al., 2023b), Llama 2: Open Foundation and Fine-Tuned Chat Models (Touvron et al., 2023a) and Falcon LLM (Almazrouei et al., 2023; Penedo et al., 2023).

2.6.7 Transfer Learning

The ability of pre-trained language models to efficiently transfer general knowledge to specific downstream tasks has significantly impacted natural language processing (NLP) research. In the past, we had to invest time and resources in training language models from scratch. However, with the availability of publicly accessible checkpoints for large language models, fine-tuning them has become more time and cost-efficient. This advancement improved the accessibility to research and made the research pipeline much faster.

2.7 Automatic Evaluation Metrics

Measuring the performance of text generation is a tricky task and is not as easy as with standard discriminative tasks such as classification tasks or sentiment analysis or named entity recognition. Evaluating sequence-to-sequence tasks is difficult because there is no definite answer to what is a good summary. When it comes to text summarisation, a single document can have multiple summaries that emphasise different content and use varied linguistic styles. Any of the available metrics is not able to generate an objective evaluation of the system summaries. However, assessing Natural Language Understanding (NLU) systems may be easier and straightforward. There exist different effective measures, such as accuracy, widely used in binary classification tasks.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.18}$$

Where
- TP (True Positive): A true positive is when the model correctly predicts a positive class instance as positive.
- TN (True Negative): A true negative is when the model correctly predicts a negative class instance as negative.
- FP (False Positive): A false positive is when the model incorrectly predicts a negative class instance as positive. It is also known as a Type I error.
- FN (False Negative): A false negative occurs when the model incorrectly predicts a positive class instance as negative. It is also known as a Type II error.

We can go further by calculating precision , recall and F1 score in the case of unbalanced datasets (see Equations 2.19, 2.20 and 2.21)

$$Precision = \frac{TP}{TP + FP} \tag{2.19}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.20}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2.21}$$

However, in the context of ROUGE, Recall means how much of the reference summary is the system summary recovering or capturing. To get a good quantitative value, we can actually compute the precision and recall using the overlap of words between the two summaries.

$$\text{Recall} = \frac{\text{Number of overlapping words}}{\text{Total words in reference summary}} \tag{2.22}$$

This measure counts the number of overlapping n-grams and word sequences between the system-generated summary and the gold standard summary. The downside of this metric is that it fails to indicate the effectiveness of the remaining words in the system summary. While a machine-generated summary (system summary) may be excessively long, including every word from the reference summary, a significant portion of these words may be irrelevant, resulting in an unnecessarily wordy summary.

The importance of precision becomes apparent in this context. It refers to the extent to which the system summary contains relevant and necessary information.

Precision is measured as follows:

$$\text{Precision} = \frac{\text{Number of overlapping words}}{\text{Total words in system summary}} \tag{2.23}$$

It is always best to compute both the Precision and Recall and then report the F1-Measure, which is the harmonic mean of the recall and precision. If the summaries are forced to be concise, then we could consider using just the Recall since precision is of less concern in this scenario.

2.7.1 Automatic Evaluation of Summarisation

Evaluating system-generated summaries is a very challenging task due to the subjectiveness of the process (Fiszman et al., 2009) since there is no apparent ideal summary. This makes human evaluation, which directly assesses the expectations of real users, the most reliable approach. Unfortunately, manual evaluation is usually very labour-intensive and costly (Hirao et al., 2007), leading researchers to propose different automatic metrics that replicate human assessments. An appropriate metric for summarisation has to rate the relevancy of the chosen material and may also assess the smoothness of the summary. Evaluating abstractive texts presents further complexities over evaluating extractive summaries because the generated summary may include words not present in the source text. Despite these difficulties, there are many automatic metrics that can be used to quickly contrast different summarisation approaches and give an impression of how good a summary is or at least to rank different system-generated summaries. The main features of automatic evaluation are: quick, low cost but inaccurate. The main criteria used to evaluate text summarisation are: fluency, informativeness, focus, coherence, clarity and non-redundancy. Furthermore, we need to make sure that a generated summary has a good readability score. One of the mostly used metrics to evaluate summarisation, are *ROUGE* metric variants (Kiyoumarsi, 2015). However, recently, there has been a surge in the use of model embedding-based metrics and statistical-based metrics.

2.7.2 Evaluation Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004), is a metric used for evaluating automatic summarisation of texts as well as machine translation[15]. It measures the number of overlapping textual units such as n-grams or word sequences, between the generated summary and a set of gold reference summaries (supports multi references) (typically human-produced). Rouge uses recall, precision and F1 score to measure the performance.

There are different variants of ROUGE: ROUGE-N, ROUGE-S, ROUGE-SU, ROUGE-LSum, and ROUGE-L. We will try to define them one by one:

- **ROUGE-N** – measures unigram, bigram, trigram and higher order n-gram overlap between the system summary and reference summary. ROUGE-N (Rouge-1, ROUGE-2, ROUGE-3 ...) can be thought of as the granularity of texts being compared between the system summaries and reference summaries. For example, ROUGE-1 refers to the overlap of unigrams between the system summary and reference summary. ROUGE2 refers to the overlap of bigrams between the system and reference summaries. ROUGE-N is a recall-based metric that measures n-gram overlap between the system and reference summaries as follows in equations 2.24 and 2.25:

$$\text{Rouge-N} = \frac{\sum_{i=1}^{N} \min(\text{count}_{\text{match}}(n_i), \text{count}_{\text{ref}}(n_i))}{\sum_{i=1}^{N} \text{count}_{\text{ref}}(n_i)} \tag{2.24}$$

  Where ‘N’ represents the length of the n-grams being considered. The most common values for N are 1 (Rouge-1 or unigram), 2 (Rouge-2 or bigram), and sometimes 3 (Rouge-3 or trigram).

$$\text{Rouge-N} = \frac{\text{COUNT MATCH}}{\text{COUNT REF}} \tag{2.25}$$

  Here, “COUNT MATCH” is the Number of overlapping n-grams between the generated text and reference text and “COUNT REF” is the total number of n-grams in reference text.

- **ROUGE-L** – measures the longest matching/common sequence of words using LCS. An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence-level word order. Since it automatically includes the longest in-sequence common n-grams, you do not

[15]Rouge Github Repository: `https://github.com/google-research/google-research/tree/master/rouge`

need a predefined n-gram length. It eliminates the need to explicitly specify the n-gram length like its counterpart, ROUGE-N, does. ROUGE-L (summary level LCS) is calculated as follows in the equation 2.26.

$$\text{Rouge-L} = \frac{\text{Length of LCS between generated text and reference text}}{\text{Length of reference text}} \tag{2.26}$$

Where the “Length of LCS” refers to the length of the longest subsequence that appears in both the generated text and the reference text. It is not necessary to specify the value of N in this case since Rouge-L is based on subsequences rather than n-grams.

- **ROUGE-S**, also known as Skip-Bigram, is an evaluation metric used to assess the quality of system-generated text in comparison to reference text. It aims to capture the similarity between the skip-bigrams in the generated text and those in the reference text. This can also be called skip-gram co-occurrence. Skip-bigrams are pairs of words where one word is skipped, allowing for more flexible matching compared to traditional consecutive bigrams. As an example, for the phrase “cat in the hat” the skip bigrams would be “cat in, cat the, cat hat, in the, in hat, the hat”. Rouge S is calculated as follows in equation 2.27.

$$\text{Rouge-S} = \frac{\sum_{n\in\text{Skip-bigrams}} \min(\text{count}_{\text{match}}(n), \text{count}_{\text{ref}}(n))}{\sum_{n\in\text{Skip-bigrams}} \text{count}_{\text{ref}}(n)} \tag{2.27}$$

  Where Skip-bigrams represent the set of all skip-bigrams in the generated text.

  And $\text{count}_{\text{match}}(n)$ is the number of times the skip-bigram n appears in both the generated and reference texts.

  And $\text{count}_{\text{ref}}(n)$ is the number of times the skip-bigram n appears in the reference text.

- **ROUGE-SU4** (skip-gram with unigrams) is an evaluation metric that is part of the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) family of metrics. It is a method where bigrams with a maximum gap of four words between them are considered matching bigrams between the evaluated and model summary. Although ROUGE takes into account contextual information, it remains at the word-level, which means we either regard different types of the same word as different or we need to apply stemming or lemmatisation to remove this effect. To summarise, in ROUGE-SU4, ‘SU’ stands for Skip-Bigram Unigram, and ‘4’ indicates that the metric considers skip-bigrams of length up to 4 (i.e. 1 to 4 words). We can say that ROUGE-SU4 considers the skip-bigram and unigram overlap between the generated text and the reference text.

One of the best implementations of Rouge score is the Java Rouge (JRouge2.0)[16] package (Ganesan, 2018). This package includes different stemmers and enables the evaluation of summaries in several languages. *ROUGE* is the most used and accurate metric in assessing text summarisation, especially the extractive one (Liu et al., 2008). However, it still (with other gram-based metrics such as *METEOR*) has some limitations. In fact, ngram metrics focus on word matching rather than other quality aspects such as fluency, grammaticality and coherence. Furthermore, given the fact that summarisation is a tricky subjective task, rouge metrics were designed to be used with multiple reference summaries per input. So, Rouge may also be not useful in the case where we have only one reference summary per input. (See (Schluter, 2017) for an in-depth discussion of many of ROUGE's limits)

METEOR (Banerjee et al., 2005) (Metric for Evaluation of Translation with Explicit Ordering) computes an alignment between candidate and reference sentences by mapping unigrams in the generated summary to 0 or 1 unigrams in the reference, based on stemming, synonyms, and paraphrastic matches. Precision and recall are computed and reported as a harmonic mean. In other terms the metric is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. METEOR is different from BLEU and ROUGE in the sense that METEOR takes into account the position of words when computing similarities between a candidate and a reference sentences which makes the comparison stricter. There are different cases that are considered as matches: *Exact*: the word from the candidate exactly matches the word from the reference sentence, *Stem*: a stemmed word (for example, walk of the word walked) matches the word from reference sentence and *Synonym*: the word from a candidate sentence is a synonym for the word from the reference sentence. METEOR is calculated as follows in equations 2.28 and 2.29:

$$\text{METEOR} = \frac{\text{Match}_{\text{exact}} \times \text{penalty}_{\text{stem}} \times \text{penalty}_{\text{frag}}}{(\text{len}_{\text{ref}} + \alpha \times \text{penalty}_{\text{stem}}) \times (1 - \beta \times \text{penalty}_{\text{frag}})} \tag{2.28}$$

$$\text{METEOR} = (1 - \alpha) \times \text{Precision} + \alpha \times \text{Recall} \times \text{Fragmentation Penalty} \tag{2.29}$$

CIDEr (Vedantam et al., 2015) (Consensus-based Image Description Evaluation) is another measure that evaluates the consensus of a candidate sentence to a given set of reference statements. It computes 1–4-gram co-occurrences between the candidate and reference texts, down-weighting common n-grams and calculating cosine similarity between the n-grams of the candidate and reference texts. CIDEr is defined to measure the grammaticality, saliency, and accuracy of a candidate sentence.

BERTScore (Zhang et al., 2020b) is an embedding-based evaluation metric that computes similarity scores by aligning generated and reference summaries on a token level[17]. Token alignments are computed greedily to maximise the cosine

[16] `https://github.com/kavgan/ROUGE-2.0`

[17] Github repo: `https://github.com/Tiiiger/bert_score`

similarity between contextualised token embeddings from the BERT transformer. BERTscore could be employed using different model checkpoints. Following the definition presented by Zhang et al. (2020b) the calculation could be defined as follows: Using a given reference sentence x $= \langle x_1, \ldots, x_k \rangle$ and a candidate sentence $\hat{x} = \langle \hat{x}_1, \ldots, \hat{x}_l \rangle$, Bertscore utilises contextual embeddings to represent the tokens, then calculate the matching by using cosine similarity between both sequences (system and reference) (see Equation 2.30 for cosine similarity). This matching can be optionally weighted with inverse document frequency scores. The computation process is depicted in Figure 2.8 for better understanding. Given two vector sequences x and $\hat{x}$, BERTScore is computed as follows: recall: equation 2.31, precision: equation 2.32 and F1 score in equation 2.33.

Note: The cosine similarity ranges between -1 and 1, with a value of 1 indicating that the two vectors are identical, a value of 0 indicating that the two vectors are orthogonal, and a value of -1 indicating that the two vectors are in opposite directions.

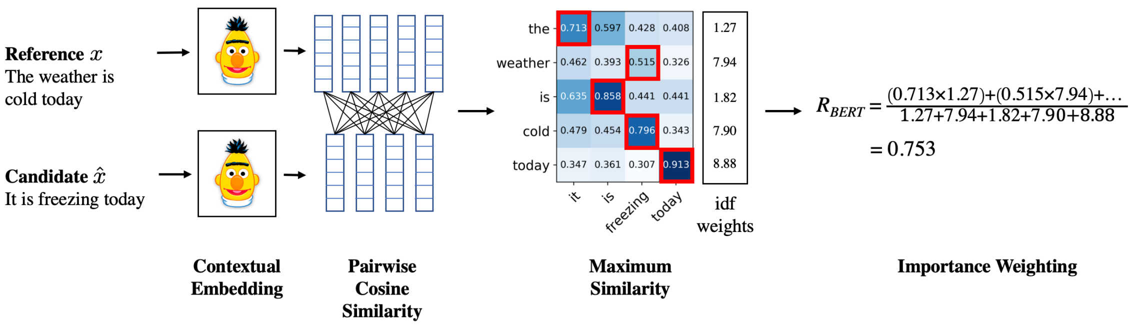


Figure 2.8: Demonstration of the computation of the recall metric of BERTscore. Source (Zhang et al., 2020b)

$$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \, \|B\|_2} \tag{2.30}$$

$$\text{Recall}_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j \tag{2.31}$$

$$\text{Precision}_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^\top \hat{x}_j \tag{2.32}$$

Then the F1-score is obtained:

$$\text{F1}_{\text{BERT}} = \frac{2 \times \text{Precision}_{\text{BERT}} \times \text{Recall}_{\text{BERT}}}{\text{Precision}_{\text{BERT}} + \text{Recall}_{\text{BERT}}} \tag{2.33}$$

Bleurt (Sellam et al., 2020) is a Transfer Learning Based Metric for Natural Language Generation. It takes a pair of sentences as input: one candidate and several references, and it returns a score that indicates to what extent the candidate is fluent and conveys the meaning of the reference. Bleurt could generate negative results. Therefore we can not use it to assess if the system summary is "bad" or "good". The best use of BLEURT should be used to compare systems on the same task and rank them. BLEURT has pre-trained checkpoints for base and large models. Bleurt is available on Github[18] and could be calculated as follows in equation 2.34:

$$\text{BLEURT} = \frac{1}{N}\sum_{i=1}^{N} \text{similarity}(\text{gen}_i, \text{ref}_i) \tag{2.34}$$

Moverscore (Zhao et al., 2019) is a semantic based evaluation metric used for summarisation, text generation, machine translation... Moverscore uses the Earth Mover Distance to compute the semantic distance by comparing two sets of embeddings (system and reference text) (See Equation 2.35). By default, moverscore uses English BERT to measure the similarity of two sentences.

$$\text{MoverScore} = \frac{1}{N}\sum_{i=1}^{N} \text{EMD}(\text{gen}_i, \text{ref}_i) \tag{2.35}$$

Frugal score: is a technique that allows us to obtain a budget-friendly version of any expensive NLG metric while preserving most of the original performance (Eddine et al., 2021) . Authors claim that experiments on summarisation and translation tasks demonstrate that the FrugalScore versions of BERTScore and MoverScore have nearly the exact correlation with human judgments as the originals but run up to 24 times faster and use 35 times fewer parameters. On average, across all metrics, tasks, and variants studied, FrugalScore preserved 96.8% of performance power while decreasing speed requirements and parameter usage considerably (Eddine et al., 2021). Frugal score is available on github[19].

BaryScore (Colombo et al., 2021b): This newly proposed metric is very adapted to the task of Natural Language Generation (NLG). It is a multi-layered metric which works based on pre-trained contextual representations. It is comparable to MoverScore as it combines the output from multiple layers of Bert before calculating a similarity score. By interpreting the layer output from deep contextual embeddings as a probability distribution instead of a vector embedding, BaryScore aggregates these outputs through Wasserstein space topology (see Figure 2.9). The original implementation of Baryscore[20] is heavily based on the Optimal Transport for signal, image processing and machine learning python library[21] (Flamary et al., 2021). The Baryscore uses the 2D free support Wasserstein barycenters of distributions introduced by the POT library [22]

[18] https://github.com/google-research/bleurt

[19] Frugalscore Github repo: https://github.com/moussaKam/FrugalScore

[20] https://github.com/PierreColombo/nlg_eval_via_simi_measures

[21] https://pypi.org/project/POT/

[22] https://pythonot.github.io/auto_examples/barycenters/plot_free_support_barycenter.html

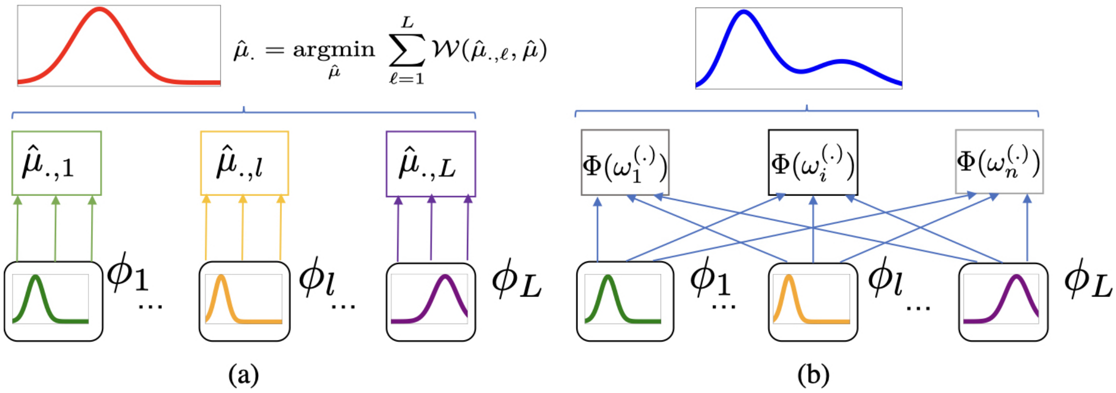


Figure 2.9: BaryScore (left) vs MoverScore (right). source (Colombo et al., 2021b)

Depthscore (Staerman et al., 2021) is a metric based on pre-trained contextualised representations that use a single layer of Bert to embed both the candidate C and the reference R and obtain discrete probability measures (Staerman et al., 2021). This statistical metric was tested on Data2text and Summarisation. The similarity score is computed using the pseudo metric $DR_{p,\varepsilon}(\hat{\mu}^C_{.,l}, \hat{\mu}^R_{.,l})$ introduced by Staerman et al. (2021).($\hat{\mu}^C_{.,l}, \hat{\mu}^R_{.,l}$) are the discrete probability measure of the candidate and reference sentences, respectively. Figure 2.10 illustrates an example of Depth score.

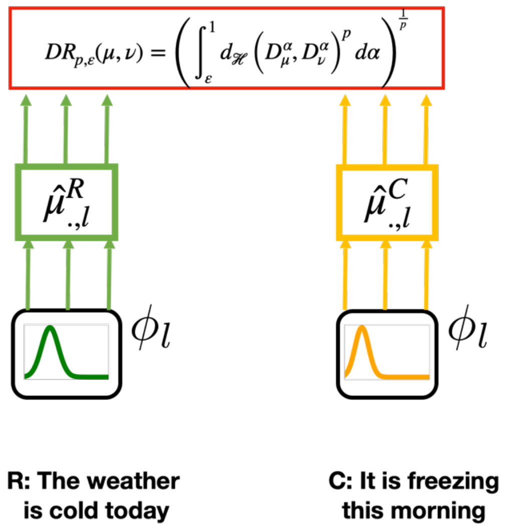


Figure 2.10: Depth Score. source (Staerman et al., 2021)

infoLM (Colombo et al., 2021a) is a metric that is adjustable and can apply various standards using distinct information metrics. This metric has gone through testing in Data2text and summarisation . It makes use of a pre-trained language model (PLM) to generate a discrete probability distribution over the vocabulary for an input sentence S, where it is masked at position i. The second component of InfoLM involves a measure of information which calculates the distance/divergence between the combined distributions (candidate C and reference R). InfoLM could be implemented using different mathematical measures of distance or divergence.

Finally, there are other metrics used for summarisation evaluation like edit distance-based metrics such as WER, TER, PER and ITER (Sai et al., 2022).

2.7.3 Human Evaluation

Human assessment of summaries usually involves creating a list of queries/questions posed to a group of human judges. These questions can evaluate different qualitative characteristics like coherence and fluency of the summary, or they can measure how much information has been maintained from the original article. Typically, an absolute numerical system, such as a 1-5 scale, is used for this kind of analysis (El-Haj, 2012); however, preference-based reviews that ask participants to pick between two summaries outputs are also a common practice in human evaluation for text summarisation (Louviere et al., 2015). The human evaluation may be accurate and significant if the sample is large enough, but it is expensive and time-consuming.

2.8 Shared Tasks

In this section, we will present the last three financial summarisation shared tasks that presented the latest research in the field of financial narrative summarisation.

2.8.1 FNS Shared Task 2020

El-Haj et al. (2020d) discusses the results and discoveries of the Financial Narrative Summarisation (FNS 2020) shared task focusing on summarising UK annual reports. FNP-FNS 2020 (El-Haj et al., 2020a) hosted this shared task with one key objective to utilise either abstractive or extractive summarisation approaches and techniques to generate automatic summaries for UK financial annual reports. FNS summarisation is the first of its kind to target financial annual records. The data used in this shared task was obtained from publicly available UK annual reports published by firms listed on the London Stock Exchange (LSE). A total of 24 systems were submitted from 9 different groups. Furthermore, there were two baseline and two topline summarisers to help judge and compare the outcomes of the participants.

2.8.2 FNS Shared Task 2021

Zmandar et al. (2021c) reports the outcomes and discoveries of the Financial Narrative Summarisation Shared Task 2021, which concentrated on condensing UK yearly reports. This task was organised as a component of the Financial Narrative Processing Workshop in 2021 El-Haj et al. (2021a). It included one primary job, which is to utilise either abstractive or extractive automatic summarisers to summarise long documents related to UK financial annual documents. This shared task is the second to be focused on financial records. The data for this shared task was obtained from publicly available UK annual reports published by companies listed on the London Stock Exchange. In all, 10 systems from 5 different teams participated in this shared task. The task also presented two baseline and two topline summarisers to assess the results of the participating teams and compare them to existing state-of-the-art systems. The participating systems used a variety of techniques and methods ranging from fine-tuning pre-trained transformers to using high-performing deep-learning models and word embeddings. In addition, the participating teams used methods to investigate the hierarchy of the annual reports to detect structure and extract the narrative sections to identify the parts in the report from which the gold summaries were extracted. Most of the applied techniques were extractive since the dataset is highly extractive.

2.8.3 FNS Shared Task 2022

El-Haj et al. (2022c) outlines the findings of the Financial Narrative Summarisation Shared Task (2022) regarding UK, Greek, and Spanish annual reports. Hosted as part of the Financial Narrative Processing 2022 Workshop (FNP 2022 Workshop in Marseille), this task has been running since 2020 and has been a component of the Financial Narrative Processing (FNP) workshop series since 2019. It requires abstractive or extractive automatic summarisers to reduce lengthy documents into summaries for UK, Greek and Spanish financial annual reports. The data used in the task was gathered from yearly reports that are available to the public from companies listed on UK, Greek and Spanish stock exchanges. In total, 14 summarisation systems by 7 different teams were submitted to FNS 2022. The various systems involved in the shared task employed many strategies and approaches, varying from tuning pre-trained transformers to incorporating high-performance deep-learning models and word embeddings. Moreover, the participating teams utilised methods to explore the organisation of annual reports in order to spot patterns and separate narrative portions so they could pinpoint which sections of the report should be used for extracting gold summaries. Since the data is highly extractive, most techniques applied were extractive. The used metric was the ROUGE-2 F measure for each language. The systems are ranked according to the final score, which is weighted as follows: English (50%), Spanish (25%) and Greek (25%).

2.9 Summary of the Chapter

In conclusion, this background section has provided a comprehensive overview of the existing literature and foundational concepts relevant to text summarisation and natural language processing. We have explored the historical milestones of text summarisation, delved into the types and different techniques used in text summarisation, and identified the gaps in current knowledge regarding financial text summarisation. We have also explored automatic evaluation metrics and presented different theoretical concepts related to transformer models and financial narratives in different reporting contexts. As we conclude this background chapter, we will move forward to the next chapters in order to answer the research questions and fill the gap in the field of multilingual financial text summarisation.

Chapter 3

Methodology

3.1 Overview of the Methodology

This chapter outlines the methodology employed in the design of the research process explored in this thesis, detailing the experimental setups, methods, research process, corpora used, data collection methods, data analysis techniques, ethical considerations, validation of findings, and study limitations. This high-level methodology chapter aims to provide a high-level overview of the research process and the interconnections between chapters, serving as a reference point for the next chapters. Each section will describe the experimental conditions and settings, highlighting the diversity of approaches and the rationale behind these choices. The methodology described here aims to ensure clarity across the various experiments despite the different tasks involved.

3.2 Experimental Design and Setup

This section details the experimental setups and conditions under which the research was conducted. This includes a detailed description of the text types, languages studied, summarisation techniques, and the rationale behind the chosen configurations. In general, the experiments conducted are designed to explore the performance and applicability of different summarisation techniques under different conditions. The key setups are described in the following subsections.

3.2.1 Text Types and Sources

To evaluate the effectiveness of summarisation techniques, we used different types of textual data: financial reports (long and medium size), preliminary earnings announcements, financial result reports, and news articles. These text types were chosen due to their differing structures, different lengths and content characteristics. News articles are characterised by their diverse topics and structured reporting style, which is highly abstractive. The headlines could be used as gold-standard summaries.

Otherwise, news articles could be manually labelled using short abstracts. Annual financial reports are noted for their technical jargon and specific informational focus. They are longer and more complex to analyse. They are labelled using extracted sections from the original text. Financial result reports are concise and contains different technical terms and numbers aiming to explain the financial performance of the firm. Section 5.2 (English Financial Narrative Summarisation Dataset), Section 5.3 (Arabic Financial Corpora), and section 5.4 (French Financial Narrative Dataset) will give detailed descriptions of the datasets and text types used in the experiments.

3.2.2 Multilingual Considerations

The experiments span three languages: English, Arabic, and French. This selection allows for a comprehensive analysis of summarisation techniques across different linguistic contexts and different vocabulary ranges.

- English: is a widely used and researched language in the field of financial natural language processing, with extensive resources and datasets available and a long track record of open source corpora. Furthermore, English is the official language of financial markets, whether in terms of reporting, analysis or communication.
- Arabic: presents unique challenges compared to other languages such as script directionality (right to left), morphological complexity, special alphabet and rich inflectional structure. Arabic is the official language of financial reporting in the Middle Eastern markets.
- French: a Latin language that offers insights into the adaptability of summarisation techniques to different grammatical and syntactic structures. French is widely used for financial reporting in several countries such as France, Belgium, Switzerland and Canada.

3.2.3 Summarisation Techniques and Tools

Two main approaches to summarisation were investigated: extractive, abstractive and hybrid summarisation.

- **Extractive summarisation**: It selects and rearranges existing sentences from the original source text to create a summary. Extractive methods aim to identify and rank key sentences based on their relevance and informativeness. Some of the basic approaches used are TextRank, LexRank, and neural network-based approaches that effectively identify key sentences or phrases in an input text. Extractive summarisation is easier to implement and faster to compute compared to other techniques but may result in summaries that are less coherent or overly verbose. The preprocessing techniques may be described as text normalisation, input tokenisation, and annotation. The evaluation is mainly conducted using automated quantitative metrics such as n-gram-based and embedding-based metrics. Extractive summarisation will be used mainly in chapters 6 and 8.

- **Abstractive summarisation**: It generates novel sentences to capture the main ideas of the source text and rewrite it in different words. The emergence of abstractive summarisation came with the evolution of open source sequence to sequence models and encoder-decoder models. Abstractive summarisation comes with some challenges such as the need for higher computational memory and extensive calculation memory. The same preprocessing steps are applied to ensure consistency. The methodologies used for abstractive summarisation are based on reinforcement learning techniques or the training of transformer-based models (e.g., BERTSUM, T5, BART, mT5). These models are fine-tuned to improve their ability to understand and rephrase content from diverse sources. For the evaluation metrics, human judgment criteria and readability scores are the benchmarks for text generation evaluation. Furthermore, embedding-based quantitative metrics, such as Frugalscore, could be used. Abstractive summarisation will be used mainly in chapter 7.

- **Hybrid summarisation**: This technique combines both the extractive and abstractive methodology. It can be either combining extractive and abstractive parts or it can consist in extracting and then rephrasing some parts.

3.2.4 Rationale Behind Experimental Configurations

The diverse experimental configurations were chosen to address several key research questions:

1. Language-Specific challenges:
   - How does language affect the performance of summarisation models?
   - What preprocessing steps are necessary to handle linguistic variations across English, Arabic, and French?

2. Impact of text type:
   - How do summarisation techniques perform on structurally different texts like news articles versus annual financial reports?
   - Are certain techniques better suited to specific text types?

3. Summarisation technique efficiency:
   - How do abstractive and extractive summarisation methods compare in terms of quality, coherence, and informativeness?
   - What are the trade-offs between computational complexity and summary quality for each method?

3.3 Data Collection and Preprocessing

A crucial component of the methodology involves a detailed description of financial corpora and datasets employed in the research. This section gives an overview of the corpora and datasets used in the experiments. The datasets were picked to pretrain models and test summarisation methods on various text types and languages. These diverse datasets play a fundamental role in enabling investigations into long document extractive summarisation of English financial annual reports, abstractive summarisation of Arabic financial news, and summarisation of medium-size French financial reports. The selection of corpora and datasets is critical to ensure a comprehensive evaluation.

3.3.1 Data Sources and Collection Methods

Financial reports are often technical, with a focus on financial metrics, company performance, and future perspectives. Our datasets are divided into three languages: **English** UK annual reports from FTSE100, **Arabic** financial annual reports from leading Middle Eastern companies, **French** financial statements from companies listed in CAC 40 and CAC 60 indexes. Section 5.2, 5.4 and 5.3 will give more details about the descriptions of the data sources and collection methods.

3.3.2 Data Preprocessing and Annotation

Effective data preprocessing is critical to the success of summarisation models. The corpora and datasets underwent extensive preprocessing and cleaning to ensure consistency and compatibility with the summarisation models. The preprocessing steps included:

1. Text normalisation and cleansing: it aims to standardise the text by removing inconsistencies and ensuring uniformity by removing special characters, numbers, and formatting inconsistencies such as unnecessary whitespace. Furthermore, text normalisation consists of handling language-specific issues, such as removing diacritics in Arabic where appropriate.

2. Tokenisation: this is an important step in the summarisation pipeline. It consists of splitting text into sentences and words using language-specific tokenisers that are trained separately to accurately split text (e.g., using Stanford NLP for English, Farasa for Arabic, and Spacy for French.).

3. Annotation: it aims to create reference summaries and key sentence annotations for training, validation and evaluation. For abstractive summarisation, human-written summaries are always used as reference summaries. For extractive summarisation, key sentences or phrases were annotated to create ground truth summaries. Annotation could be manual or using a rule-based automated methodology.

3.4 Machine Learning and Transformer Models

3.4.1 Model Selection and Justification

Several machine learning techniques are used, particularly transformer encoder-decoder models like T5, BART, Pegasus and LongFormer. The selection of these models is based on their architecture, allowing them to encode the input through the encoder and decode the output through the decoder, which makes them suitable for generative tasks. Section 6.5.1, Section 7.4.1 and section 8.4 will give more background details on the models used.

3.4.2 Training and Fine-Tuning Procedures

Training and fine-tuning transformer models involve several key steps, including dataset preparation, model selection, hyperparameter tuning, and evaluation. They also involve specifics such as the datasets used, training duration, hardware utilised, and hyperparameter settings. The training duration can vary significantly depending on the model size, dataset size, and hardware used. Typical training runs can last from a few hours (in the case of finetuning) to several days (in the case of pretraining). For hardware utilised, we may use a single GPU for smaller models and datasets and multiple GPUs for larger models and datasets to speed up training and parallelism. Otherwise, we may use TPUs (Tensor Processing Units), which can be used to train large models efficiently. For Hyperparameter settings, there are several possibilities. However, the main hyperparameters to optimise are learning rate, batch size, epochs and optimiser. In section 6.6.5, section 7.4.5 and section 8.5, we will go for a detailed explanation of the training and fine-tuning procedures.

3.4.3 Performance Evaluation

Another significant contribution is the rigorous evaluation of summarisation techniques across multiple dimensions. This comprehensive evaluation includes:

1. Multilingual performance assessment: we conducted extensive experiments to evaluate summarisation performance in English, Arabic, and French. This assessment highlights the strengths and limitations of various techniques and variations of metrics in different linguistic contexts. We performed also some stress tests over some evaluation metrics.

2. Domain-Specific analysis: we evaluated summarisation models on diverse text types, such as news articles and financial reports. This analysis provides insights into how different domains affect the effectiveness and reliability of summarisation methods.

3. Evaluation metrics and human judgment: we usually used a combination of automated metrics (e.g., ROUGE, Bertscore, Frugalscore, Meteor) and human judgment to assess summary quality. Conducting human evaluation aims assess the coherence, readability, and informativeness of the summaries. This dual approach

ensures a holistic evaluation, capturing both quantitative and qualitative aspects of summarisation performance.

3.5 Addressing Challenges and Considerations

3.5.1 Challenges in Data Handling

Several challenges were encountered during data preparation such as:

- Availability of financial data: financial data is generally proprietary data owned by companies and financial institutions. Therefore, it is very difficult to have access to qualitative data and we always have some constraints with copyrights.
- Language-specific nuances: Arabic text requires special handling for right-to-left script and diacritics. French text requires careful handling of accents and special characters.
- Data quality and consistency: ensuring the quality and consistency of summaries across different datasets was crucial for reliable evaluation. The PDF to text process is very complex and may generate noise and some conversion errors. Even the OCR process may not be efficient with scanned files. These challenges are mentioned in detail in sections 7.4.4 and 5.4.4.6. Addressing inconsistencies in annotation and formatting is crucial to maintaining dataset integrity.
- Domain-specific vocabulary: financial reports contained special terminology that required domain-specific preprocessing and annotation techniques to ensure that models could handle technical jargon and industry-specific terms effectively. By addressing these challenges and carefully selecting and preprocessing the datasets, a robust foundation was established for evaluating the summarisation techniques under different conditions.

3.5.2 Model Implementation Challenges

Implementing and training NLP models on long financial documents involves several challenges, such as handling long document input or ensuring multilingual capabilities or memory issues. The first challenge with model implementation is handling the input of long financial documents. In fact, many language models have limitations on the length of input they can process, which can be problematic for tasks involving lengthy documents containing excessive information, such as UK financial annual reports, making it difficult for models to identify and extract relevant data. Techniques such as chunking the text into smaller segments and using hierarchical models can help. Additionally, techniques such as using the most important parts as input may also be a relevant solution, allowing the model to maintain context over longer texts. Section 6.5.1 will explain how we can bypass this issue. In general, It is not always efficient to use large language models as black-box functions for summarisation. So, we often need some input data handling. Another problem with data is the use of multi-reference gold standards in most of the summarisation datasets. Section 6.6.2 and section 6.6.3 will explain how to handle this issue. Another challenge with model implementation is computational resources. In fact, training large models or finetuning them on custom use cases requires substantial computational power and memory, which can be a barrier for research (see section 6.10). In this case, we may use distributed computing or cloud resources, and model parallelism may help manage memory requirements.

3.6 Research Contributions and Advancements

The primary research contributions of this thesis are:

- Development of advanced summarisation techniques: both extractive and abstractive summarisation methods are explored, catering to different text types and languages. We explored both supervised and unsupervised techniques. Machine learning and transformers have contributed to achieving important advancements in the field. Section 6.5.1 (Summarisation Techniques) and section 7.4.1 (Pretraining a Language Model) and section 8.7 will give more details on the developed techniques.

- Evaluation of summarisation performance: performance is evaluated across various corpora, including long and medium-sized financial annual reports, in multiple languages (English, Arabic, and French). We integrate both automatic and human evaluations. Also, we integrate statistical significance tests in section 6.8 and section 8.10.4.

- Analysis of language and domain variability: the impact of language and domain on summarisation effectiveness is assessed, providing insights into the generalisability of the methods. Section 7.5.5 and section 8.10 will give an example of language and domain variability analysis.

- Cross-Linguistic adaptations: adapting summarisation techniques to handle linguistic variations across English, Arabic, and French. This includes addressing language-specific challenges such as morphological complexity in Arabic and syntactic variations in French.

- Domain-Specific customisations: tailoring summarisation methods to different text domains, including news articles and financial reports. This involves incorporating domain knowledge and terminology to improve the relevance and precision of summaries.

3.7 Conclusion of the Methodological Framework

The methodological framework outlined in this section provides a structured approach to the research conducted in this thesis. It ensures consistency and reliability across experiments and serves as a foundation for the development, evaluation, and analysis of summarisation techniques. This framework integrates various components, including data preprocessing, model training, and evaluation, to comprehensively address the research questions. In a nutshell, this methodology chapter is a high-level abstraction methodology which lays the groundwork for the comprehensive exploration of the different experimental conditions in the next chapters, highlighting their important role in shaping the research outcomes.

Chapter 4

Financial Word Embeddings

4.1 Introduction

Natural Language Processing is increasingly being applied to analyse the text of many different types of financial documents. For many tasks, it has been shown that standard language models and tools need to be adapted to the financial domain in order to represent domain-specific vocabulary, styles and meanings properly. In this chapter, we describe the creation of novel financial word embeddings for two languages (English and French), that we will need in the following chapters of the thesis. The main novel contributions of this chapter are the multilingual financial word embeddings themselves. The developed financial word embeddings will be used later in section 6.5.5 and section 8.8 to develop a neural network based financial summarisers.

4.2 Theoretical Background

Word embeddings are a collection of techniques for mapping words or sentences to high-dimensional vector space. Developing embedding vectors aims to generate word vector representations useful for various tasks like text classification, text summarisation, or sentiment analysis. The process is considered self supervised since the model learns to make predictions about some aspect of its input data without requiring explicit labels or annotations. Adjusting word embeddings involves fine-tuning several hyperparameters (such as the dimensionality of word embedding vectors), similar to any machine learning model. Word embedding models have demonstrated greater efficiency compared to bag-of-word models or one-hot-encoding schemes. In fact, they represent tokens as dense and lower-dimensional vectors (Abdalla et al., 2020). This allows for the relative positioning of words to reflect their contextual usage and other semantic aspects, such as degrees of similarity (e.g. the word ‘company’ could be similar to ‘group’). Furthermore, the trained word embedding model can allow us to visualise the vocabulary and get the vector of each word in the corpus.

4.3 Related Work

Many tasks require embeddings of domain-specific vocabulary that models pre-trained on a generic corpus may not be able to capture. Standard word2vec models are not able to assign vectors to out-of-vocabulary words and instead use a default vector that reduces their predictive value. The less generic the content of the subsequent text modelling task, the more preferable training a task-oriented model is. However, quality word embeddings are data-hungry and require informative documents containing hundreds of millions of words. Financial documents include words that appear in any general-purpose pre-trained word embedding, such as 'GloVe'. However, the usage of these words will be different; therefore, the link in the vector space should also be different. The domain-specific vocabulary used in financial disclosures is different from the 'general' language. Loughran et al. (2011) showed that the meaning of words can change substantially in a financial context. In fact, the context of a word tells you what type of words tend to occur near that specific word. Context is important in finance as this will give meaning to each word embedding. For example, corporate earnings releases use nuanced language not fully reflected in 'GloVe' vectors pre-trained on Wikipedia articles. Moreover, when working with industry-specific documents, the vocabulary or its usage may change over time as new technologies or products emerge. For all these reasons, working on training custom word embedding for the financial domain would have added value.

4.4 Motivation of Training Word Embeddings

Token mapping: if we tokenise our input text, we need to convert the token ids and the vocabulary set into a numerical representation that the machine can handle. We can assign each word in the text a unique number or perform one-hot encoding. This is not efficient either computationally or in terms of memory. We will end up with very large dictionaries of words and their corresponding mapped numbers. Despite its advantages as an initial solution, the mapping method has a major disadvantage: it does not take into account the meaning of words inside their context and inside a sentence. Therefore, semantically similar words will be mapped randomly to vectors that are completely dissimilar. This was a big constraint for language modelling and led to the development of word embedding algorithms using several neural network methods, which vectorise tokens in such a way that similar words will be close (have a high cosine similarity close to one). These networks are based on the idea that "the word is defined by its context" (Firth, 1957), and so similar words should have similar vectorial representation. Word embeddings refers to a set of algorithms in which words are represented using N dimension vector within a previously determined vector space. Every word is assigned to one vector, and these vectors' values are acquired through a black-box algorithm process.

4.5 Word Embeddings Usecases

We can use word embeddings in two ways:

- Pre-trained embeddings: learned from a generic large corpus such as Wikipedia or Google News. Popular open-source options include Stanford's GloVe (Pennington et al., 2014) and spaCy's built-in vectors. GloVe is an unsupervised algorithm developed at the Stanford NLP lab that learns vector representations for words from aggregated global word co-occurrence statistics. Vectors pretrained on the following web-scale sources are available:
  -Common Crawl with 42 billion or 840 billion tokens and a vocabulary of 1.9 million or 2.2 million tokens.
  -Wikipedia 2014 + Gigaword with 6B tokens and a vocabulary of 400K tokens.
- Domain specific trained embeddings: training domain-specific embeddings using documents that reflect a domain of interest (eg. Finance, healthcare). In fact, many tasks require embeddings of domain-specific vocabulary that models pre-trained on a generic corpus may not be able to capture. Standard word2vec models are not able to assign vectors to out-of-vocabulary words and instead use a default vector that reduces their predictive value.

4.6 Training Choices

To create word embeddings, we always need to choose an embedding method. In order to build the financial word embeddings, we used the word2vec model introduced by Mikolov et al. (2013) from Google. Word2vec is developed by using two-layer neural networks. The selection of the Word2Vec model is justified based on its capability as a robust unsupervised word embedding technique. The usefulness of Word2vec is its ability to group the vectors of similar words together in vector space and detect similarities mathematically. We can train word2vec models using two open-source libraries:

- **Gensim** library: it is more robust and faster because Gensim is compiled into C language and is based on the first paper published by Google about the word2vec algorithm.

- **Keras** library: an open-source library developed by Google that comes with the TensorFlow project. Keras allows to train word embeddings using neural networks.

To implement the word2vec model, we used the python Gensim[1] library implementation. The two variants of the Word2vec model are:

[1] https://radimrehurek.com/gensim/

- **CBOW**: The continuous-bag-of-words model predicts the target centre word using the average of the context word vectors as input so that their order does not matter. CBOW trains faster and tends to be slightly more accurate for frequent terms but pays less attention to infrequent words.
- **SG**: The skip-gram model uses the target word to predict the words surrounding a given input word. It works well with small datasets and finds good representations even for rare words or phrases. The skip-gram model implicitly factorises a word-context matrix that contains the pointwise mutual information of the respective word and context pairs (Goldberg et al., 2014).

4.6.1 Training Setup

One epoch of word embedding training takes approximately 10 minutes on a modern `4-core i7` processor and `40 Gb` RAM. The training speed can be significantly improved by using parallel training on multiple-CPU machines. The main choices that impact the performance of the model are:

- **architecture**: skip-gram (slower, better for infrequent words) vs CBOW (faster). The default training algorithm is CBOW(0) or skipgram(1).
- **alpha**: The initial learning rate - (0.01, 0.05)
- **sample**: The threshold for configuring which higher-frequency words are randomly down-sampled. Highly influential - (0, 1e-5).
- **size**: The number of dimensions of the embeddings (Dimensionality of the word vectors): Default value is 100. 300 is the dimension we recommended for this task.
- **window**: The maximum distance between a sentence's current and predicted word. For skip-gram, usually around 10, for CBOW around 5.
- **min_count**: The minimum count of words to consider when training the model. Words with occurrences less than this count will be ignored. The default for min_count is 5.
- **workers**: The number of partitions during training and the default workers is 3.

The parameters we used to train word2vec model are shown in Table 4.1:

4.6.2 Training Process

In this section, we explain the training process of domain-specific embeddings (in English and French) using annual financial reports from UK firms and annual financial reports from French firms. We will first describe how we pre-processed the data

sg	min_count	window	size	sample
1	3	2	300	6e-5
alpha	negative	workers	epochs	—
0.05	20	16	15	—

Table 4.1: Word2Vec Training Parameters

for this task, then demonstrate how the skip-gram architecture outlined in the first section works, and finally visualise the results. It is better to separate the training into 3 distinctive steps for clarity and monitoring.

Step 1: 'Word2Vec()': In this first step, we should set up the parameters of the model one-by-one.

Step 2: '.build_vocab()': In this step, we build the vocabulary from a list of sentences and thus initialised the model. 'min_count' and 'sample' parameters have great influence over the performance of a model.

Step 3: '.train()': Finally, trains the model. The loggings here are mainly useful for monitoring, making sure that no threads are executed instantaneously.

Pre-processing typically involves phrase detection, that is, the identification of tokens that are commonly used together and should receive a single vector representation. Figure 4.1 shows how word embeddings are trained starting from a corpus.

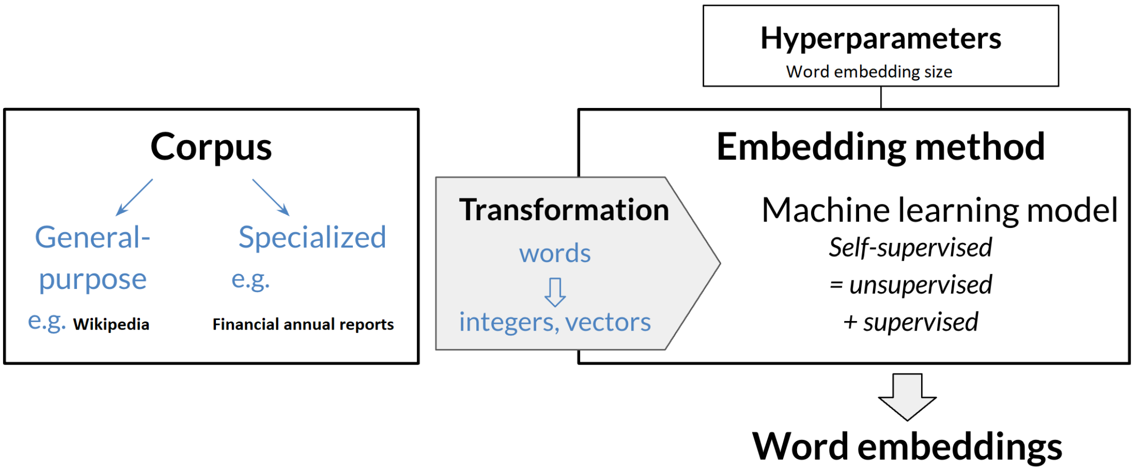


Figure 4.1: Word embedding training process

In order to employ the word2vec model for financial purposes and ensure accurate data representation, we start by performing the pre-processing using NLTK[2] library.

[2] `https://pypi.org/project/nltk/`

We deleted non-alphanumeric values and replaced some special characters by their equivalent (e.g. “m” is replaced with “million”). Additionally, all words were converted to lowercase. Moreover, we extracted tokenised sentences of the dataset using the NLTK tokeniser and created a vocabulary of the training dataset in the form of a dictionary where keys are words and values are the number of occurrences. The tokenised sentences were passed as input to the word2vec tool from the ‘Gensim’ library, which produced the word vectors as output. For efficiency purposes, we limited our vocabulary size to 20,000 by selecting only the most frequent words. Furthermore, we set a maximum sentence length of 60 words.

4.7 Datasets

We used two different datasets to train our two-word embeddings.

- **FNS + Annual Reports Key Sections Corpora**[3]: FNS dataset is UK annual report dataset in English from the financial summarisation shared task 2020 (El-Haj, 2019; El-Haj et al., 2020e; Zmandar et al., 2021d). The dataset is composed of 3,000 annual reports from UK firms.
  Annual Reports Key Sections Corpora is a Plain text content extracted from an initial sample of 31,464 annual reports published between January 2002 and December 2017 by firms listed on the London Stock Exchange (LSE) (El-Haj et al., 2020b).
- **COFIF** (Daudert et al., 2019): A Corpus of Financial Reports in French Language[4]. It contains over 188 million tokens in 2,655 reports from French-listed companies in the CAC40 (the French stock market index).

4.8 Visualisation of Embeddings

An example of a 2D plot of French embedding is shown in Figure 4.2. The plot is realised using the t-SNE dimensionality reduction algorithm (Maaten et al., 2008). The algorithm takes a query word (“Personne” in this case) and its list of most similar words. The plots are plotted using the Seaborn library, and the algorithm is implemented through the Scikit Learn library. From this plot above, we can see that when encoding a word in a two-dimensional plot, similar words tend to be found next to each other.

4.9 New Frontiers

Although pretrained Word2vec and GloVe embeddings capture more semantic information than the bag-of-words approach and allow better results on different

[3] https://doi.org/10.17635/lancaster/researchdata/271
[4] https://github.com/CoFiF/Corpus

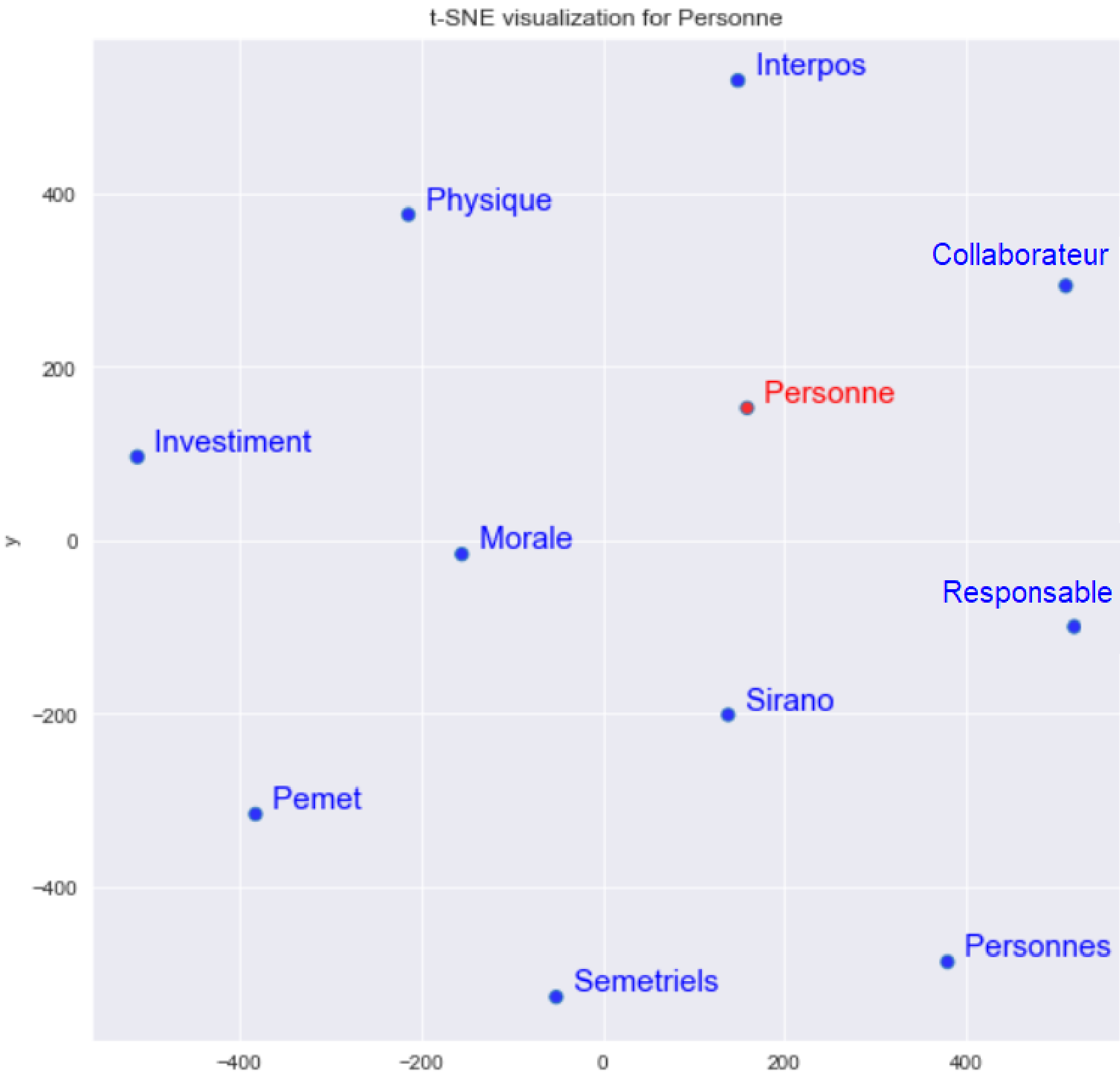


Figure 4.2: T-SNE 2D plot for the French word "Personne"

NLP tasks, they are unable to differentiate between context-specific usages. To address unsolved problems like polysemy, several models have emerged that build on the attention mechanism designed to learn more contextualised word embeddings. In December 2017, Vaswani et al. (2017) published their seminal paper, "Attention Is All You Need", describing their work at Google Research and Google Brain, presenting the original transformer model. Since then, the use of bidirectional language models that process text both left-to-right and right-to-left for a richer context representation has emerged, and the use of semi-supervised pretraining on a large generic corpus to learn universal language aspects in the form of embedding that can be used for fine-tuning for specific tasks. This is what we will test in chapter 6 and 8.

4.10 Summary

In this chapter, we have reported on training financial word embeddings in two languages[5]. We have outlined the various steps and hyperparameters employed in the process, as well as provided a theoretical background on word embeddings. The trained word embeddings will be used in financial summarisation in Section 6.5.5 and Section 8.8.

[5] https://github.com/UCREL/multilingual-financial-word-embeddings

Chapter 5

Trilingual Financial Corpus

Within this chapter, we will present four different corpora used in this thesis. In Section 5.2, we will present the English financial narrative summarisation dataset that will be used in Chapter 6. In Section 5.3, we will present two Arabic corpora that we collected for this thesis work. And finally, in Section 5.4, we present the French Financial Narrative Summarisation Corpus: CoFiF Plus, which was published at the LREC2022 conference and will be used in Chapter 8.

5.1 Introduction

As the number of digital financial reports increases, new methods for gathering and organising the data into an easy-to-understand form are essential. Investors needed help to read through thousands of documents with hundreds of pages. In addition, the investment management business is becoming more quantitative and systematic and looking for new markets and languages. Therefore, summaries for financial reports could be an effective way to communicate written information with different stakeholders, as they condense the document down to only its most essential elements and also direct readers' attention to relevant parts in a document, hence making the equity research process more straightforward. In France, Price Waterhouse Coopers (PWC) publishes a yearly guide named "Financial Communication Framework And Practice" by PWC PwC France and Maghreb (PwC France, 2019; PwC France, 2020; PwC France, 2021; PwC France, 2022) covering the French Financial market. The goal of text summarisation is to take lengthy documents and reduce them to concise summaries that retain the key points and meaning. This technique can be approached in an abstractive or extractive way, as well as for individual or multiple documents (Miller, 2019). In order to enhance the work on financial narrative summarisation and develop novel financial summarisation approaches, datasets containing original texts as well as very well-selected gold standard summaries are required, whether we are aiming to use extractive or abstractive methods. However, financial summarisation is an uncommon task among public datasets (Abdaljalil et al., 2021), due to the specificity of this application field where data is not shared by companies and accessing data can be very costly. Hence, the next sections present different financial corpora in three languages.

5.2 English Financial Narrative Summarisation Dataset

5.2.1 Financial Text Summarisation Corpus

An annual report is a document that businesses in the UK and abroad create yearly or Biannually to provide a thorough overview of their activities and financial performance over the past year. This type of document is intended for shareholders and other company's stakeholders. One of the most crucial points to training financial text summarisers is to collect datasets. Different sources of financial text could be financial social media posts, financial data providers, financial news agencies, annual reports and earnings calls. The currently available data is the English dataset of annual reports that we used in the Financial Narrative Summarisation shared task (FNS 2021[1], FNS 2022[2]) where we focus on annual reports produced by UK firms listed on the London Stock Exchange (LSE) and circulated in PDF format in English. (Zmandar et al., 2021e; El-Haj, 2019; El-Haj et al., 2019b; El-Haj et al., 2020f; El-Haj et al., 2021c). The dataset contains approximately 4,000 UK annual reports covering the period between 2002 and 2017 (El-Haj et al., 2019b). The average length of an annual report is approximately 50,000 words. The dataset includes 9,873 gold standard summaries. The dataset is divided into *training*, *testing* and *validation* sets providing both the full text of each annual report along with gold-standard summaries. The dataset is randomly split into training (75%), testing and validation (25%). Table 5.1 shows the dataset details stats per split. Full details of the labelling process of this dataset and the extraction process are available here (El-Haj et al., 2019b; El Haj et al., 2018; El-Haj et al., 2019b)

Data Split	Train	Validate	Test	Total
Report full text	3,000	363	500	3,863
Report sections	60,794	9,247	12,089	82,130
Gold summaries	9,873	1,250	1,673	12,796

Table 5.1: FNS Shared Task Dataset

5.2.2 Dataset Statistics

Table 5.2 shows some detailed statistics about the FNS summarisation dataset. We report the average/median word and sentence counts in the dataset.

[1] https://wp.lancs.ac.uk/cfie/fns2021/

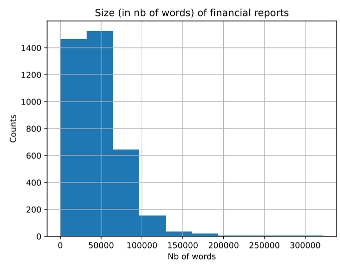
[2] https://wp.lancs.ac.uk/cfie/fns2022/

Data Type	Train	Val	Test	Total
# Report	3,000	363	500	3,863
# Gold summaries	9,873	1,250	1,673	12,796
# words / report (average / median)	43,843/37,495	60,324.68/58,412	62,039.84/54,947	47,747.04/40,259
# sentences / report (average / median)	1,904.03/1,592.5	2,611.76/2,543.0	2,700.89/2,410	2,073.68/1,740
# words / summary (average / median)	1,072.82/752	1,193.91/828	1,168.91/867	1,097.21/774
# sentences/summary (average / median)	50.17/37	56.68/40.5	55.25/41	51.47/38

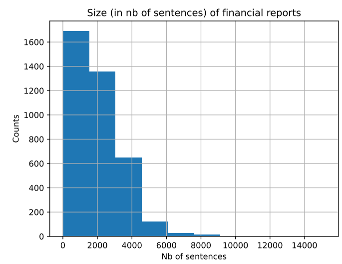
Table 5.2: English Financial Narrative Summarisation Dataset Statistics

The Figures 5.1 5.2 5.3 5.4 show the distribution of the number (Nb) of sentences and words in the financial reports and the respective summaries (counts in the y-axis means the number of reports). We perform this analysis on the training, validation; testing splits and on all the corpus.

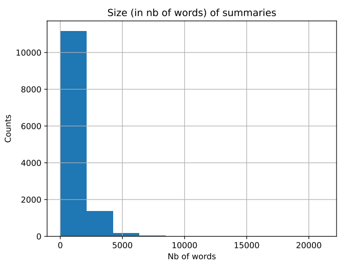
Figure 5.5 shows the number of annual reports that have n gold standard summaries. n could range from one to seven. To get the total number of gold standard summaries, we had to calculate the sum of the number of summaries multiplied by the number of reports (top of the bar boxplots). However, we can see that all the reports have at least three gold standard summaries. This is the main idea behind creating this dataset. Since we do not have a clear definition of what is a good summary, we opt for a multi-labelled dataset. This results in a more objectively labelled corpus. However, it also introduces a significant bias towards evaluation, as our system summary will be compared to more than one gold standard summary.



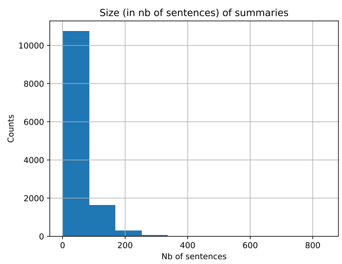
(a) Word Distribution for reports



(b) Sentence Distribution for reports

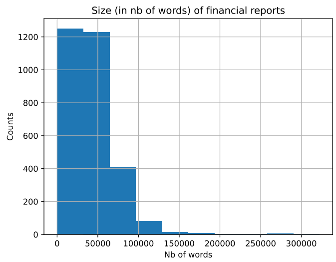


(c) Word Distribution for Summaries

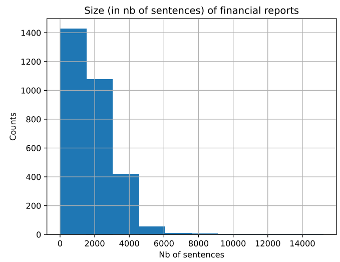


(d) Sentence distribution for summaries

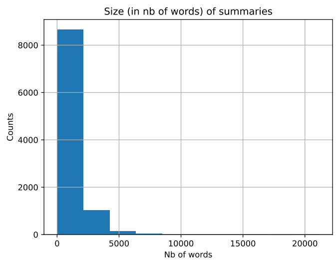
Figure 5.1: Distribution of number of word / sentence in the annual reports and their gold standard summaries (Plots on all the dataset)



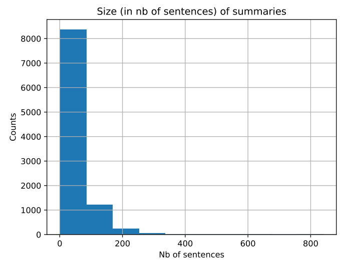
(a) Word Distribution for reports



(b) Sentence Distribution for reports

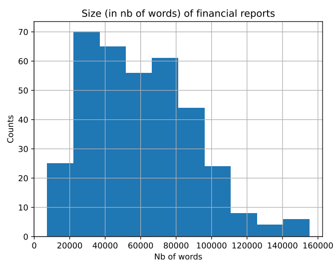


(c) Word Distribution for Summaries

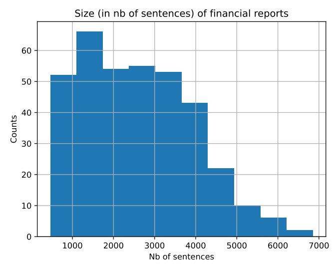


(d) Sentence distribution for summaries

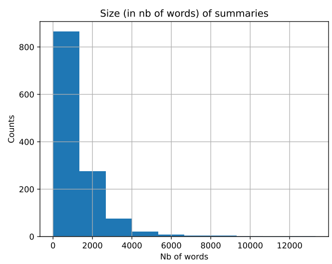
Figure 5.2: Distribution of number of word / sentence in the annual reports and their gold standard summaries (Training Split)



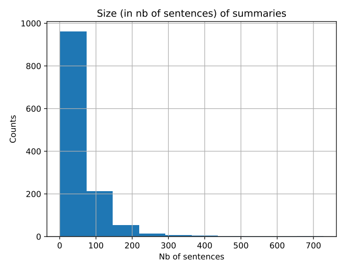
(a) Word Distribution for reports



(b) Sentence Distribution for reports

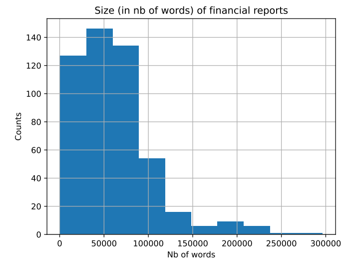


(c) Word Distribution for Summaries

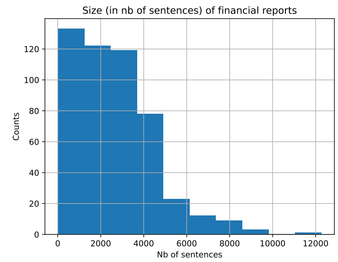


(d) Sentence distribution for summaries

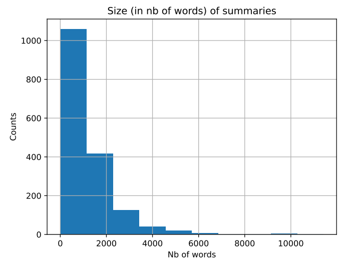
Figure 5.3: Distribution of number of word / sentence in the annual reports and their gold standard summaries (Validation Split)



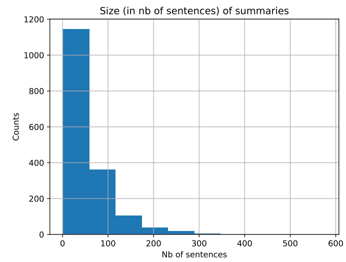
(a) Word Distribution for reports



(b) Sentence Distribution for reports

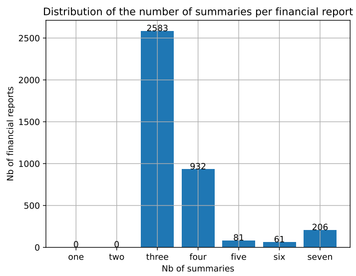


(c) Word Distribution for Summaries

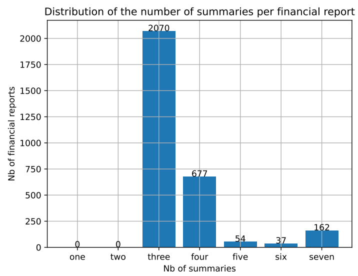


(d) Sentence distribution for summaries

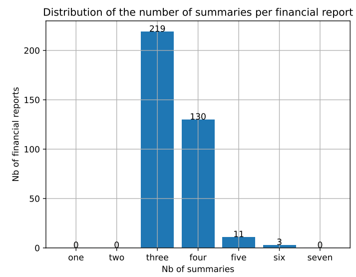
Figure 5.4: Distribution of number of word / sentence in the annual reports and their gold standard summaries (Test Split)



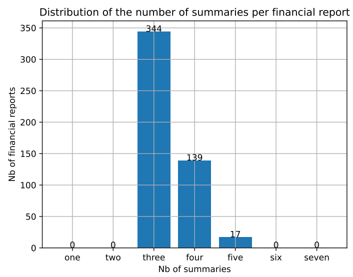
(a) All the dataset



(b) Train Split



(c) Validation Split



(d) Test Split

Figure 5.5: Distribution of the number of gold standards per annual report (All/ Training / Validation/ Test)

5.2.3 Data Analysis of the English FNS Dataset

To gain a deeper comprehension of the FNS dataset, we conducted a series of data analysis processes and visualisations, as depicted below.

Unigrams: The Unigrams plot represents key financial technical Unigrams used in the corpus. Among the most represented unigrams, we can mention: group, report, financial, million, market, growth, plc. (see figure 5.6a)

Bigrams: The Bigrams plot represents key financial technical Bigrams used in the corpus. Among the most represented bigrams, we can mention: annual report, plc annual, report accounts, financial statements, per share, strategic report, chief executive, operating profit, corporate governance, balance sheet, cash flow, and consolidated statements. This confirms the technical aspects of financial reports and their difference from other general-purpose textual data. (see figure 5.6b)

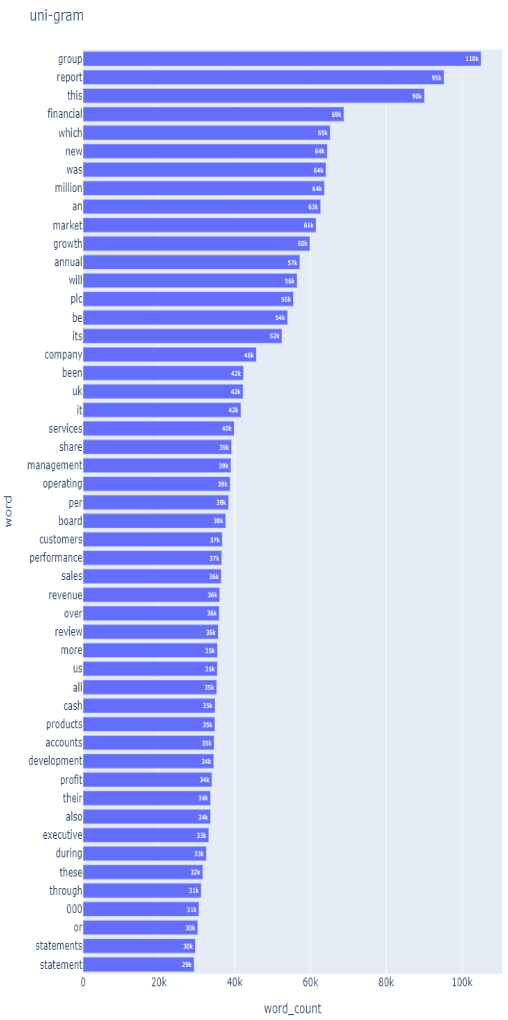
Treemap: A TreeMap is an effective data visualisation technique employed to represent hierarchical data as nested rectangles. It provides an overview of the data (unigram) frequency. We use the treemap in our data exploration and visualisations to enhance the understanding and analysis of textual data by identifying the mostly used unigrams in the corpus. It allowed us to quickly assess the relative significance of different financial terms and identify the dominant words used. By examining the treemap, we discovered that the most recurrent words in our English dataset are: group, report, this, which, new, was, financial, million, market, growth, annual, plc, company, services, share and management. (see figure 5.8)

Word Cloud: Using the word cloud is useful since it provides a quick overview of the most frequent or important words in the dataset. Some NLP studies used Word Cloud to explore the dataset (Coppersmith et al., 2014; Sornlertlamvanich, 2014). The word cloud displays the importance of words in the text based on frequency, typically through font size or font or colour. Also, it helps to detect outliers that are apparent from the visualisation. In the case of figure 5.9, we did not do any preprocessing such as stemming or removing stop words.

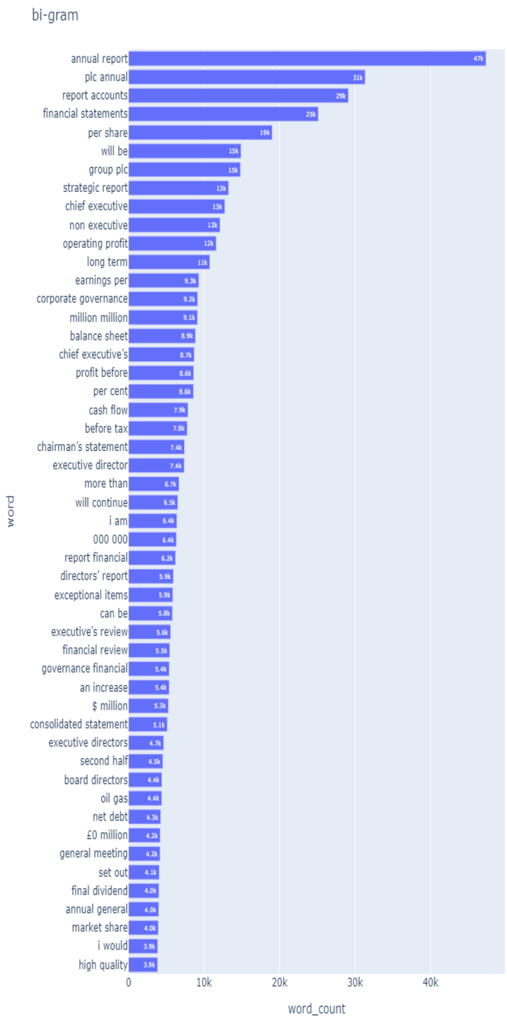
Word Distribution: Figure 5.7 shows the word count distribution. We have a gaussian (normal) distribution of the word count appearance in the whole FNS corpus.

5.2.4 FNS Dataset Representation

Every annual report has its own individual identifier, which is used to match the full text of the report with its gold standards. The annual report and its gold summary are formatted in the format reportID.txt and reportID_summaryID.txt, respectively as in 17.txt and 17_1.txt, 17_2.txt, 17_3.txt . The dataset is presented following the structure in figure 5.10.



(a) unigrams distribution



(b) bigrams distribution

Figure 5.6: unigrams and bigrams distribution for the English dataset

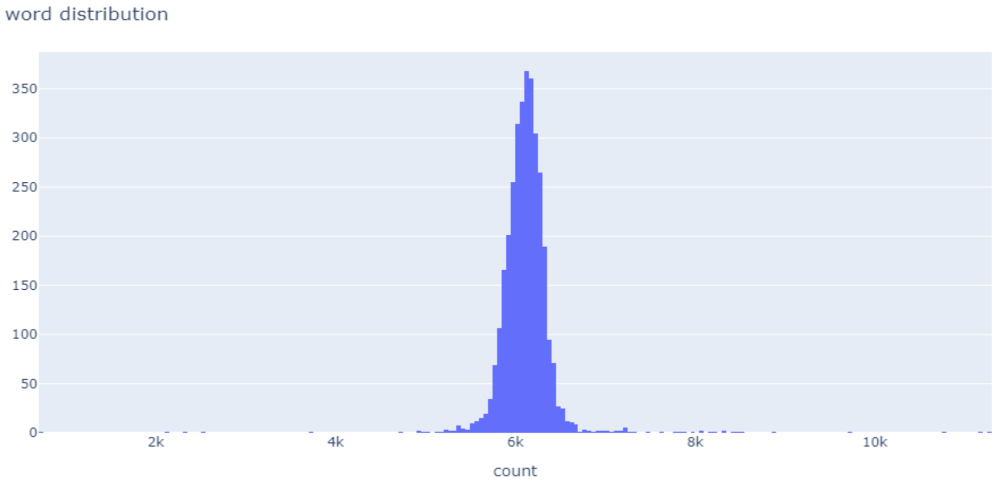


Figure 5.7: Distribution Analysis of English Word Frequencies

Figure 5.8: Treemap of FNS dataset

Figure 5.9: Word cloud of FNS dataset

```
Financial Narrative Dataset
|------training
        |------annual_reports
        |------gold_summaries
|------validation
        |------annual_reports
        |------gold_summaries
|------testing
        |------annual_reports
```

Figure 5.10: English Dataset Structure

5.3 Arabic Financial Corpora

5.3.1 Large Corpus of Arabic Financial Reports

Training a transformer model needs a large corpus in plain text because of the large number of parameters in the model's architecture. There is no available public financial corpus covering financial statements in Arabic. Hence, we had to create the training corpus ourselves from scratch. We have aggregated two corpora of different orders of magnitude to train the models. In this subchapter, we present the Arabic financial corpus employed in this research which consists of a collection of financial annual reports in Arabic from reputable financial sources.

5.3.1.1 Financial Corpus Acquisition

In this section, we describe in detail our approach to collecting large-scale financial text in Arabic. The task is challenging, as financial reports are not readily available or centralised in one location. The acquisition of the Arabic financial corpus involved various steps in collecting the relevant data from a credible source. We employed web scraping techniques to construct the data. The corpus was meticulously curated to ensure the representativeness of the Arabic financial domain.

Data Acquisition: We collected several types of financial documents from different Middle Eastern financial markets, such as auditor reports, earning announcements, accounting documents, quarterly reports (Q1, Q2, Q3, Q4), annual financial reports and management board reports. The Middle Eastern financial narrative reports come in PDF file format. There is no predefined structure, and files contain pictures and a lot of colours. A total of 30,000 PDF files were collected. The total size of PDF files collected is around 25 GBs. We focused on major stock exchanges in the Middle East to collect our corpus. Our data is collected from the following Arab markets: Kingdom of Saudi Arabia (KSA)[3] (TASI[4], NOMU[5]), United Arab Emirates(UAE) (Dubai[6] and Abu Dhabi[7]), Kuwait[8], Oman[9] , Qatar[10] and Bahrain[11]. The corpus is a diverse set of documents from different sectors and covers several categories. We have more than 35 categories (E.g. financial services, banking, insurance, telecommunication, oil and gas, energy, real estate, basic materials, pharmaceutical, healthcare, media and entertainment, food production, consumer

[3]KSA exchange: https://www.saudiexchange.sa/

[4]Tadawul All Share Index. https://www.saudiexchange.sa/wps/portal/tadawul/markets/equities/indices/today?locale=en

[5]Saudi Parallel Market NOMU (Growth) parallel market https://www.saudiexchange.sa/wps/portal/tadawul/knowledge-center/about/parallel-market?locale=en

[6]Dubai Financial Market (DFM). https://www.dfm.ae/

[7]Abu Dhabi index https://www.adx.ae/

[8]Boursa Kuwait. https://www.boursakuwait.com.kw/en/

[9]Muscat Stock Exchange. https://msx.om

[10]Qatar Stock exchange. https://www.qe.com.qa/

[11]Bahrain stock exchange. https://www.bahrainbourse.com/

services and utilities). We did not include the Egyptian financial disclosures as they were not freely accessible and required a premium payment for data access. For other North African markets, such as Morocco and Tunisia, companies communicate mainly in French rather than Arabic. Table 5.3 describes the corpus in detail by providing statistics about the different indexes used in this corpus.

Index	**Tasi**	**Nomu**	**Dubai**	**AD**	**Kuwait**	**Qatar**	**Oman**	**Bahrain**
# companies	223		178	73	163	47	111	42
MKT cap	3158.57		294.83		105.98	165.39	13.00	24.60
Time range	2003-2021		2009-2021		2012-2021	2010-2021	2015-2021	2014-2021
# reports	19651		3338		3192	2454	23	536
# sectors	21		11		13	7	2	6

Table 5.3: Statistics about the financial Arabic corpus. The columns represent the different indexes used. The rows describe the number of listed companies included in the report, market caps in US billion dollars, time range of the corpus, number of reports collected and the number of sectors included in the corpus. AD stands for Abu Dhabi stock exchange.

5.3.1.2 Summary

We presented the Arabic financial corpus that will be utilised in the subsequent Chapter 7 to train our generative language models. This sub-chapter outlines the purpose of the corpus and its acquisition process. In Chapter7, we will go into the details of the cleaning and preprocessing process applied to the Arabic financial corpus to make a clean qualitative text to train a model. While this subchapter provides an overview of the dataset collection, Chapter 7 will comprehensively explain each step of converting the PDF files into texts. We will discuss the specific methodologies and software used for PDF-to-text conversion and OCR. Then, we will discuss corpus normalisation, handling of special characters, removing English words, removing 'tatweel', and diacritics, removing repeated characters, and addressing any noise or inconsistencies in the raw dataset. In total, we will explore all the techniques relevant to optimising the quality of the corpus.

5.3.2 Financial Arabic News Summarisation Dataset

The Financial Arabic News Summarisation Dataset, called "FinAraSum dataset", was inspired by the XSum dataset (Narayan et al., 2018c) and OrangeSum dataset (Kamal Eddine et al., 2021).

5.3.2.1 Data Collection Process

We followed the collection procedure described by Narayan et al. (2018c) and Kamal Eddine et al. (2021), who created Xsum and OrangeSum, both are highly abstractive datasets. We collected the newswires from "alarabiya asswaq" website[12]. The choice of this news source is motivated by the fact that it is the largest news website in Saudi Arabia and the Middle East, with 21 million monthly visitors. Alarabya has specialised financial and economic journalists writing several articles daily covering the region's financial news. They mainly use Modern Standard Arabic and European number notation. The collected dataset covers several categories: financial markets, economics, real estate, energy, economy, tourism and special stories. We collected all the available news articles covering a decade from 2012 to 2021. We decided to create our own Arabic financial news dataset to solve the issue of the need for more NLP datasets. The choice was to create a dataset adapted to abstractive summarisation, which is news headline generation. This will enable testing the efficiency of the pretrained model by testing the generative component of the model, which is itself a challenging task in NLP. Every collected JSON file is composed of a news article, a title, a date and a category. We included some examples of the dataset structure in Appendix A.1. Table 5.4 shows some examples of headlines included in the corpus. We also provide English translations.

Table 5.5 shows the number of articles in each category. The total number of articles is over 49,000, with the most represented categories being companies, economy, oil & gas and financial markets. In addition, we report the total number of words per category, the average article length, the average title length and especially the compression ratio which indicates the ratio between the length of the input document and the gold standard summary (Chen et al., 2020). We have an average compression ratio of 4.8%, which is a high level.

5.3.2.2 Statistics about the FinAraSum

Table 5.6 compares FinAraSum with previously released datasets such as CNN, DailyMail, NY Times, OrangeSum and XSum datasets. The dataset is smaller than Xsum, CNN, NYT, and Daily Mail but larger than the OrangeSum title and OrangeSum abstract. Table 5.6 shows that the article body and the title are 238.3 and 9.0 words in length on average, respectively, and they are 12.74 and 1.0 sentences in length on average, respectively. The dataset was very clean and did not require any specific post-processing. Table 5.7 shows that the dataset is more abstractive than the previously released one, making it a very challenging task for our financial pretrained model. There are 37.8% novel unigrams in the FinAraSum Gold summaries, compared with 35.76% in Xsum, 26.54% in OrangeSum title, 30.03% in OrangeSum Abstract, 16.75% in CNN, 17.03% in DailyMail, and 22.64% in NY Times. Similar results are reported for Bigrams, Trigrams and 4-grams. It can be observed that FinAraSum is more abstractive than XSum, OrangeSum, and traditional summarisation datasets.

Abstractivity measure: The percentage novel n-gram defines the percentage of new n-grams that appeared in the title and did not exist in the news article body.

[12] https://www.alarabiya.net/aswaq

Category	**Title in Arabic**	**English Translation**
companies	تسلا توقف الإنتاج في أكبر مصانعها عالمياً بسبب مشاكل الإمداد	Tesla halted production at its largest factory in the world due to supply problems.
economy	وزير الاقتصاد الياباني: من الصعب علينا التوقف عن استيراد النفط الروسي	Japanese Economy Minister: It is difficult for us to stop importing Russian oil.
Financial Markets	الإسترليني يتراجع بفعل علامات على تعثر الزخم الاقتصادي	Sterling is falling on signs of faltering economic momentum.
Oil and Gas	ارتفاع صادرات النفط السعودية إلى ٧ ملايين برميل… أعلى مستوى منذ ٢١ شهرا	Saudi oil exports rose to 7 million barrels...the highest level in 21 months.
Real estate	مبيعات المنازل الجديدة في الولايات المتحدة تسجل هبوطاً حاداً في يونيو	New home sales in the US fell sharply in June.
Special Stories	مورغان ستانلي يحذر من تراجعات كبيرة لمؤشرات البورصة الأميركية	Morgan Stanley warns of significant declines in US stock indices.
Travel and Tourism	أميركا تسجل أكبر عدد ركاب للطائرات في يوم منذ الجائحة.. ٢٫٤٠ مليون مسافر	America records the largest number of aircraft passengers in a day since the pandemic... 2.24 million passengers

Table 5.4: Examples of headlines in each category from FinAraSum.

Category	# articles(000s)	# words(000s)	Avg article	Avg title	compression %
companies	9,8	2049,859	208.51	9.34	5.923
economy	9,0	2257,193	250.30	9.19	4.994
Financial Markets	8,4	1881,032	223.00	8.85	5.174
Oil and Gas	8,7	1937,520	221.63	8.90	5.591
Real estate	4,6	1209,406	259.86	8.56	4.387
Special Stories	3,8	1234,289	323.45	9.54	3.538
Travel and Tourism	5,0	1234,534	245.24	8.45	4.578
Total	49,4	11803,833	238.32	9.00	5.103

Table 5.5: Number of financial newswires and words in each category.

Dataset	Train/Val/Test	Avg Doc Length		Avg Summary length		Vocab Size	
		words	Sentence	words	Sentence	Docs	Sum
CNN	90.3/1.22/1.09	760.50	33.98	45.70	3.58	34	89
Daily mail	197/12.15/10.40	653.33	29.33	54.65	3.86	564	180
NYT	590/32.73/32.73	800.04	35.55	45.54	2.44	1233	293
Xsum	204/11.33/11.33	431.07	19.77	23.26	1.00	399	81
Orangesum title	30.6/1.5/1.5	315.31	10.87	11.42	1.00	483	43
Orangesum Abstract	21.4/1.5/1.5	350.00	12.06	32.12	1.43	420	71
FinAraSum	44.90/1.5/2.5	238.30	12.74	9.00	1.00	11803	445

Table 5.6: Sizes (column 2) are given in thousands of documents. Document and summary lengths are in words. Vocab sizes are in thousands of tokens as reported in (Kamal Eddine et al., 2021)

Dataset	% of novel n-grams in gold summary				LEAD		
	Unigrams	Bigrams	Trigrams	4-grams	R-1	R-2	R-L
CNN	16.75	54.33	72.42	80.37	29.15	11.13	25.95
Daily mail	17.03	53.78	72.14	80.28	40.68	18.36	37.25
NYT	22.64	55.59	71.93	80.16	31.85	15.86	23.75
Xsum	35.76	83.45	95.50	98.49	16.30	1.61	11.95
Orangesum title	26.54	66.70	84.18	91.12	19.84	08.11	16.13
Orangesum Abstract	30.03	67.15	81.94	88.3	22.21	07.00	15.48
FinAraSum	37.8	73.6	89.0	95.2	18.30	07.5	14.79

Table 5.7: Degree of abstractivity of FinAraSum compared with that of other datasets, as reported in (Narayan et al., 2018c) and (Kamal Eddine et al., 2021).

ALGORITHM 1: Percentage of Novel n-grams in a file Y

Data: x: first doc (reference doc, e.g. news article), y: second doc to verify the overlap (e.g. title or summary), ngrams: number of n grams

Result: Percentage: a dictionary of the percentage of new ngrams in y for n ngrams

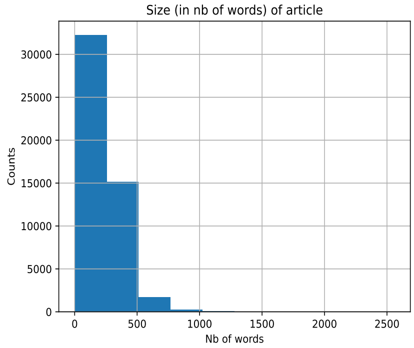
begin
  x = x.translate(str.maketrans(", ", string.punctuation)).lower()
  y = y.translate(str.maketrans(", ", string.punctuation)).lower()
  percentage = dict()
  ngrams$_x$ = $set(ngrams(x.split(), n))$
  ngrams$_y$ = $set(ngrams(y.split(), n))$
  percentage[n] = round(100*len(ngrams$_y$.$difference(ngrams_x))/len(ngrams_y), 1)$
  return *percentage*
end

Algorithm 1 shows an example pseudocode explaining how we can calculate the abstractivity percentage. Table 5.8 shows the Number of title summaries with at least n % new n-grams from the original financial news article.

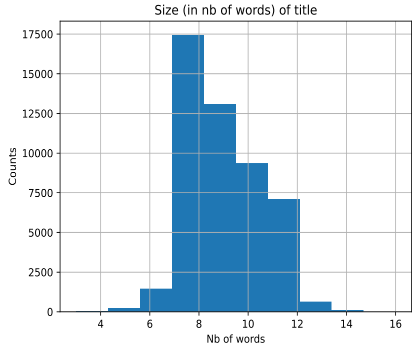
Model	**nb of title summaries with at least n % of new n-grams**						
n	**5**	**10**	**20**	**30**	**40**	**50**	**60**
unigrams	48233	47593	41626	31985	22147	15038	7109
bigrams	49460	49441	49135	48036	46388	44168	38168
3-grams	49477	49474	49344	49004	48534	47922	46196

Table 5.8: Number of title summaries with at least: n % new n-grams from the original financial news article

Finally, figures 5.11a and 5.11b describe the distribution of the size (in number of words) of the news articles and their headlines. For the headlines, we can see that most of them have between 7 and 12 words. For the articles, the majority have less than 500 words.



(a) Distribution of size (in number of words) of the articles

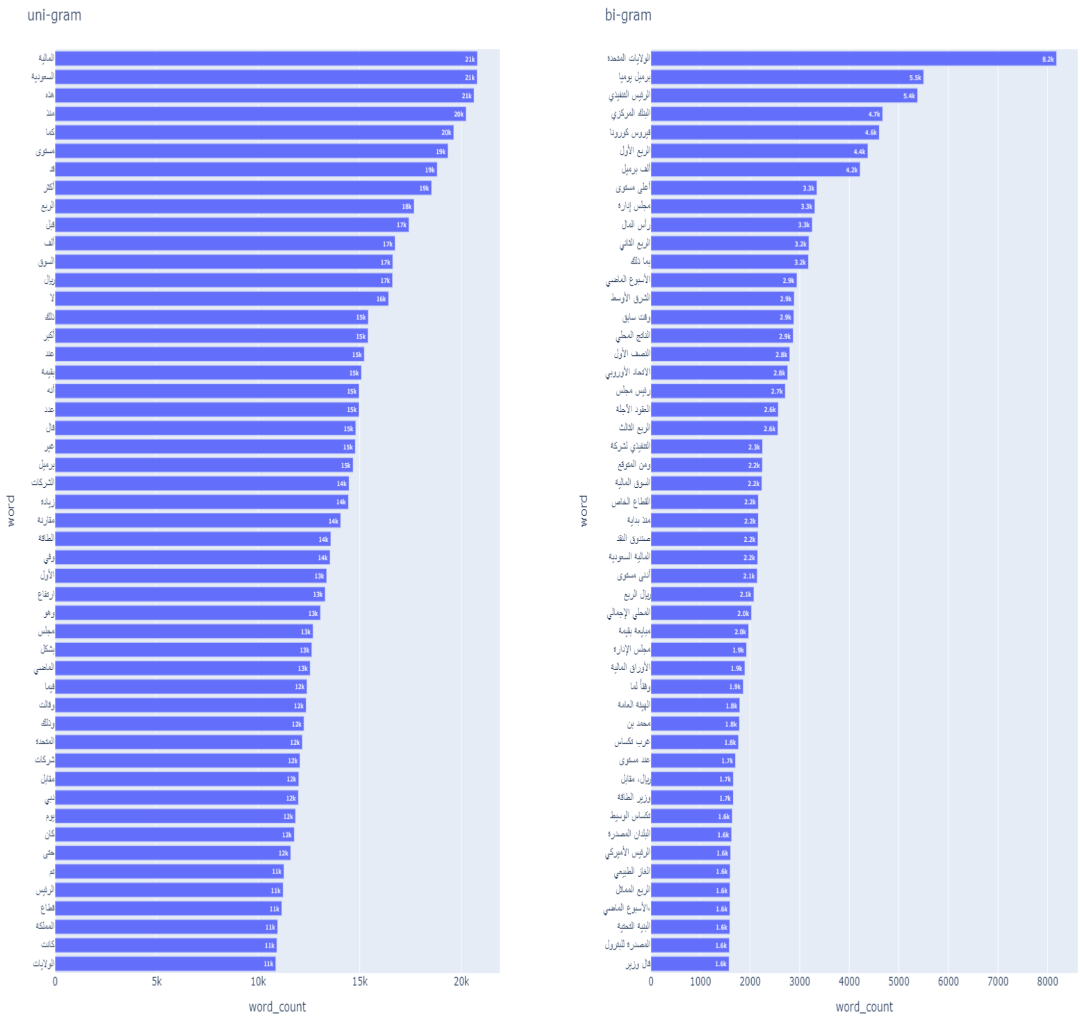


(b) Distribution of size (in number of words) of the titles

Figure 5.11: Distribution of size (in number of words) of the articles and their titles

5.3.2.3 Dataset Visualisation

The Figures 5.12a 5.12b 5.13 show the distribution of the unigrams, bigrams words and treemap respectively. Figure 5.13 plots the treemap of unigrams on the Arabic dataset.



(a) unigrams distribution

(b) bigrams distribution

Figure 5.12: unigrams and bigrams distribution for the Arabic dataset

Figure 5.13: Tree map of Arabic corpus

5.4 French Financial Narrative Summarisation Corpus: CoFiF Plus

In Section 5.2, we showed that corpora for financial narrative summarisation in English exists, and there was significant work on English financial summarisation through the FNS shared tasks series (El-Haj et al., 2020f; El-Haj et al., 2021c; Zmandar et al., 2021c; El-Haj et al., 2022c), but there is a huge lack of financial text resources in other languages such as the French language. To overcome this challenge, we have created CoFiF Plus, the first financial narrative summarisation dataset, which provides a comprehensive set of French-language financial text. This dataset was collected from PDF files of financial reports of the most capitalised companies on Euronext Paris, covering a time frame from 1995 until 2021. We describe the collection, the labelling and all the pipeline to construct this summarisation corpus. We also explore and describe the dataset by providing different statistics and quantitative and qualitative features.

5.4.1 Previously Created French Corpora

There is no research in French on summarisation within the financial domain. There are a few examples of work on French summarisation (Kamal Eddine et al., 2021; Zhou et al., 2022), however, this is not in the financial domain. This is largely due to the lack of existing corpora enabling such research.

Masson et al. (2020) created a French and dialectal French corpus specifically designed for NLP analytics related to finance, regulation and investment called "DoRe". This corpus comprises 1769 Annual Reports from 336 businesses among the most highly capitalised in France (Euronext Paris) & Belgium (Euronext Brussels), covering the period from 2009 to 2019, and it also includes MetaData associated with each company, such as its ISIN code, capitalization and sector. This corpus has been designed to be as flexible as possible so that it can be utilised for a variety of tasks concerning Economics, Finance and Regulation.
Daudert et al. (2019) presented CoFiF, the first text corpus with company reports in French. It consists of 2655 documents (annual, semestrial and trimestrial reports, Document de Reference) that have a total of 188 million words and focuses on the 60 largest French firms listed on CAC40 and CAC Next 20. The corpus covers a period of 20 years, from 1995 to 2018.

5.4.2 Corpus Creation Motivation

Financial NLP researchers working on French financial text summarisation require labelled financial annual report datasets. This includes a set of documents together with their corresponding summaries (one or more). Currently, there is no financial French summarisation dataset. There are no benchmark summaries or previously published datasets. So, the first aim of this Section is to create Cofif Plus in order to move forward with French financial automatic summarisation. This Chapter outlines the development of machine-created dataset resources.

5.4.3 Financial Communications in France

French firms are listed on the Euronext Paris[13] which is France's securities market, formerly known as the "Bourse De Paris". The French equities market is divided into two main markets as follows: the *Premier Marché* (Primary Market) where the securities are offered for the first time for investors for a fixed price, and the *Second Marché* (Secondary Market) where financial securities are traded between investors following the rule of offer and demand.

In France, there is an association that organises the job of financial communication professionals, which is the French association of financial communication professionals (Cliff[14]). Companies provide communications mainly in HTML or PDF format. In addition, XBRL[15] format became a requirement in 2022, which will facilitate mining financial narratives [16]. In terms of deadlines for filing, an annual financial report must be filed, published and disseminated to the market stakeholders no later than four months after the close of the tax/fiscal year. In France, the *Autorité des marchés financiers* (Financial Markets Authority), ensures standardisation in reporting financial results by requiring companies to follow a certain template. But it is not a form to fill like in the United States. So, every firm could use its own style while respecting the general framework. French firms can also publish an English version, but it is a translation of the French version and they have to mention that it is not the original one.

France's corporate financial information environment has the following different types of French narratives (AMF France, 2021), (PwC France, 2019; PwC France, 2020; PwC France, 2021; PwC France, 2022):

- **Annual financial reports:**

  The AMF(Autorité des marchés financiers) recommends that public firms engaged in the trading of financial securities on regulated markets disclose their annual turnover data for the previous tax year at the earliest possible date but no later than two months after its end. In order to meet the requirement to disseminate accurate and precise financial communication, the communication should present the significant events of the period as well as their impact on the accounts. It is crucial that the press-release mentions the annual accounts, turnover, annual net income and balance sheet information, particularly on equity, debt and liquidity and a management report or a chairman highlights.

- **Biannual financial reports:**

[13] https://www.euronext.com/en/markets/paris

[14] https://cliff.asso.fr/en

[15] XBRL is the open international standard for digital business reporting: https://www.xbrl.org

[16] XBRL France: https://www.xbrlfrance.org/

The AMF (Autorité des marchés financiers) recommends that listed companies on a regulated market with the AMF as the competent authority for the control of their periodic information should issue a financial report concerning the consolidated accounts for the previous six months as soon as they have been accepted by the board of directors or examined by the board of surveillance. This provides good market information about the company's activities and any significant risks and uncertainties it may have faced in the last six months. The half-year financial report includes the following: Condensed accounts or full accounts for the past semester, half-yearly activity report, and the auditors' report on the accounts appearing in the report.

- **Quarterly reports/communications:**

  Listed companies can decide to publish quarterly financial information. The AMF draws the attention of issuers to the risks associated with a complete lack of financial communication for a long period (breach of the French permanent information obligation) and recalls that this lack of information is not conducive to the proper functioning of the market. However, quarterly reports are less and less used by SMEs. This is motivated by the fact that overcommunication may lead to artificial volatility in the equity market on the announcement date. In addition, a three-month timeframe is not long enough to announce new achievements or new measures by the companies. Instead, large corporations in the cac40 are using quarter financial press releases.

- **Other communications:**

  Companies can release a Declaration of extra-financial annual performance ("DREF"), press communication, public offer announcement, bond issue announcement or profit warning. Companies can also release pro-forma financial statements/information, which must be provided as soon as transactions (carried out or planned) significantly modify the financial statements of the entities concerned. Events that may trigger these include acquisition or disposal, mergers or demergers and potentially even partial asset contribution. The pro-forma information covers and explains to an investor or shareholder the impact that such a transaction would have. This may extend to the historical financial statements of a company if this transaction had occurred at a date before its actual occurrence.

- **Document de Reference:**

  The 'reference document' is a detailed annual report published by companies listed in France. It is also known as the "document d'enregistrement universel" since the regulatory reform of 2017. This document constitutes a complete source of information on the company, its strategy, activities, governance, financial statements and many other relevant aspects. The main sections usually included in a reference document are Company Overview, Risk factors, Activities, Strategy, Corporate Governance, Legal and regulatory Information,

> Financial Statements, Information on shareholders and capital, Social and environmental responsibility, and Additional Information.
>
> The Reference document is an essential tool for investors, financial analysts, regulators, and the general public to better understand the company as a whole. It provides a complete and detailed overview of its activities, financial situation and prospects.

There is a difference between the financial reports and the 'document de reference' that should be explained. In fact, the reference document is a more complete and detailed annual report which provides the global vision and strategy of the company, going beyond financial statement information. It addresses aspects such as activities, strategy, corporate governance, social and environmental responsibility, etc. The reference document aims to offer an in-depth understanding of the business in all its aspects to help stakeholders make informed decisions. However it is a very long document with a lot of narrative and non financial sections.

On the other hand, the annual financial report focuses specifically on the company's financial performance and the key performance indicators. It focuses on financial statements, such as the balance sheet, income statement, EBITDA, and cash flows, as well as comments and details about those financial statements. The financial report presents the company's financial performance for the past year, focusing on key figures, financial trends, financial ratios and other relevant information.

In summary, the reference document is broader and encompasses all important aspects of the business, while the financial report focuses specifically on financial information. The registration document is often used as a more comprehensive communication tool for stakeholders, while the financial report is mainly intended to provide detailed financial analysis for investors and analysts.

Practically, French companies communicate their financial communications to the French financial market regulator AMF[17]. French financial communications can be found on company websites and the official financial communication data lake[18] which is managed by the "directorate of legal and administrative information". All the data comes from the Financial Markets Authority directly. A high level of regulation imposes that information should be available for the last 10 years in public.

5.4.4 Corpus Creation

There is no publicly available dataset for financial summarisation in French, which leaves large financial French markets unconsidered. Therefore, we present the first dataset for financial summarisation in the French language called CoFiF Plus, built as an extension of the corpus of financial reports in the French language (CoFiF) previously published by Daudert et al. (2019). It builds upon part of CoFiF and covers, in addition, the period 2018-2021. The final presented corpus contains 1,703 reports and 2,990 gold summaries.

[17] https://www.amf-france.org/en

[18] https://info-financiere.fr/

5.4.4.1 Corpus Description

Our criteria for selection are based on the coherence of documents published in the area of economics and finance, which can be divided into three distinct categories: annual results ("résultats annuels") that give a summary of a company's financial performance over the past year; semestrial results ("résultats semestriels") and trimestrial results ("résultats trimestriels"), which are similar to annual reports but published every six months and three months respectively. We found that most companies produce annual and semestrial reports regularly, while quarterly reports have less regular publication.

5.4.4.2 Data Selection

Our dataset is an extension of the CoFiF dataset created by Daudert et al. (2019) and follows the same design principles. We brought the dataset up to date (until the first semester of 2021). Therefore, our corpus's time span ranges between 1995 and 2021. We selected two stock indices referenced on Euronext. The first ones include the most capitalised companies: the CAC 40[19] and the second ones are the CAC next 20[20]. A total of 70 companies were selected for inclusion in our summarisation dataset. We did not include all the reports from the original CoFiF. There was a manual selection where the main criteria were the readability of the text file and the ability to extract the gold standard. Some old reports are not well formatted or contain a lot of noise. Therefore, they can not be processed using our rule-based script. The more we advance in the timeline of reports, the more we find a detectable structure, and the easier it becomes to apply the pdf2text algorithm. We ensured that we included only the reports that would help develop system summarisation models on good-quality data. In addition, we did not include "document de reference" described previously to avoid the creation of huge bias related to length or the technique of reporting and narration since "document de reference" are very long and comes with much more narrative sections rather than financial information. However, in this case, we need medium-size reports focusing on the financial performance of the company to create a well-annotated corpus.

5.4.4.3 Data Acquisition and Cleansing

We had to collect the reports for the last three years to complete the time range. The collection was mainly done by consulting the official financial communication portal[21] from the French government. One of the issues that arises while constructing such a dataset is that we cannot include the same companies in all the corpus due to the different mergers and acquisitions that occurred during the last 25 years and also due to periodic change in the composition of the two chosen indexes (due to company market cap changes). In 2020, Alstom, Teleperformance and Worldline has joined the

[19] https://en.wikipedia.org/wiki/CAC_40
[20] https://en.wikipedia.org/wiki/CAC_Next_20
[21] https://info-financiere.fr/

CAC40 index while Accor, Sodexo and TechnipFMC had left it. Another example, in 2017 Atos and STMicroelectronics joined the index and Nokia and Klepierre had left it[22]. In addition some mergers and acquisitions changes the composition of the index. For example, the French oil giant “Total” rebranded as “Total Energies” in early 2021. “Accor” also rebranded and became “AccorHotel” in 2015. “Orange S.A.” was previously “France Telecom S.A”. In May 2004, “Air France–KLM” was created by the mutually agreed merger between “Air France” and “KLM”. Thomson Multimedia changed its name to Technicolor SA in 2010, and re-branded again to Vantiva in 2022. Fnac Darty was created in 2016 following the merger between Fnac and Darty. Once we collected all the PDF files we used the pdf2text[23] python library to extract programmatically plain text from the collected PDF files. The pdf2text library adequately extracts text; however, the final result has significant noise from the poor conversion of the original structures within financial reports. Therefore, the resulting plain text files were refined using a rule-based script.

5.4.4.4 Used French Named Entity Recognition

Named Entity Recognition (NER) is the technique of obtaining information and facts from text, including names, places, organisations, phone numbers, times and dates (Ghosh, 2009). NER is a widely researched subject in NLP that figures out who or what is being discussed in a text and involves classifying mentions of named entities. These entities can be persons (PER), places (LOC), organisations (ORG), products or dates/times (DATE). There was a significant interest in developing and benchmarking French Named Entity Recognition models during the last years (Copara et al., 2020; Ortiz Suárez et al., 2020; Ortiz Suarez et al., 2022; Park, 2018; Martin et al., 2020). In this work, we will focus on one of the most used fine-tuned French NER transformers, which has more than 1 million monthly downloads from the Huggigface cloud repository. It is camembert-ner[24], a NER model fine-tuned from camemBERT[25] on the wikiner-fr dataset (170,634 sentences). The camembert-NER transformer will help us detect one of these five entities: O: Outside of a named entity, MISC: Miscellaneous entity, PER: Person’s name, ORG: Organisation and LOC: Location.

5.4.4.5 Corpus Markup and Annotation

Choice of gold standard summaries: A crucial consideration is defining what constitutes a gold standard summary of a financial report. We must consider the efficacy of extractive or abstractive methods with the goal of a readable and informative summary. A further decision is if we should include only narratives, non-narratives, or a combination (including tables). A summary of a financial

[22]Source: `https://www.bnains.org/archives/histocac/histocac.php`
[23]`https://pypi.org/project/pdftotext/`
[24]`https://huggingface.co/Jean-Baptiste/camembert-ner`
[25]`https://huggingface.co/docs/transformers/model_doc/camembert`

report is generated mainly for the shareholders and the stakeholders of the company, which needs to have a clear overview of the performance of the company. So we opted for short and concise summaries. For the French financial reports, we used the **chairman highlights**, **financial highlights** and the **general overview or perspective parts** as gold standard summaries. Therefore we will have between one and three gold standard summaries per financial report depending on the availability of the mentioned parts.

The *chairman highlight* is a concise, subjective summary of the activity during the last year, semester or quarter. It starts with words such as "a déclaré ", "Commentant", "Directeur Général", "annonce", "indiqué". Moreover, the chairman's highlights are always in the first third of the report, where the CEO comments on the company's activities and performance. It is an opportunity for the CEO to give general highlights to investors and to announce future directions and strategies. They are beneficial for investors whose main confidence in the company comes from their confidence in the managerial skills of the chairman or the CEO.

Financial highlights are generally included in the middle of the report, and they detail key financial information such as the annual turnover, EBITDA, operating income, net income, earnings per share, financial ratios, cash flows and dividends.

The *perspectives* part details the company's future plans for the next period. It gives the strategic priorities of the company and the next year's operational and financial objectives. This paragraph is essential for stakeholders to predict the company's future direction. The perspective part is always at the end of the report.

Extraction of chairman highlights as gold standard summaries: To extract the gold standards, we developed a custom heuristic rule algorithm. So let us start by explaining the idea behind using such a technique. As stated previously, French company reports come with a clear structure with recurrent patterns. This helps to find some heuristic rules to extract these summaries semi-automatically. In every French report, we will find a person's declaration. It can be either the CEO or the CFO. The common thing is that the first and last names of the person are used in addition to their function in the firm. All firms use the name of the CEO before starting his comments (e.g. Frederic Rose, Directeur general de Technicolor, a declare ...). Therefore, we will go line by line and check using the NER transformer if there is a name of a person. The chairman highlights were extracted with the help of camembert-ner[26] presented previously. The NER transformer will help us detect the name of the chairman or CEO. The Python code 5.1 shows how we can apply the French NER to find the probability that we have a person entity in an input sentence. The fine-tuned model is compelling in detecting persons in a French text.

[26] `https://huggingface.co/Jean-Baptiste/camembert-ner`

```

from transformers import AutoTokenizer
from transformers import AutoModelForTokenClassification
from transformers import pipeline

tokenizer = AutoTokenizer.from_pretrained("Jean-Baptiste/camembert-ner")
model = AutoModelForTokenClassification.from_pretrained("Jean-Baptiste/camembert-ner")

nlp = pipeline('ner', model=model, tokenizer=tokenizer, aggregation_strategy="simple") # comment

result = nlp("En 2010, nous devrons en outre nous concentrer sur  l optimisation  de nos marges sur toutes les lignes de m tiers, en am liorant le mix  d affaires  en vie, le ratio combin  en dommages et la collecte nette en gestion  d actifs . Henri de Castries, pr sident du directoire  d AXA ")

[{'end': 242,
  'entity_group': 'PER',
  'score': 0.99759406,
  'start': 224,
  'word': 'Henri de Castries'},
 {'end': 273,
  'entity_group': 'ORG',
  'score': 0.99494195,
  'start': 270,
  'word': 'AXA'}]
```

Listing 5.1: Python French NER Code

Once found, we will add another indication using an OR operator to ensure that it is not the CFO(Chief Financial Officer) or any person involved in the company's report. we manually found that the name of the CEO comes with other indications such as , "Commentant", "ces chiffres", "Directeur Général" , "a déclaré", "indiqué". This second condition avoids retrieving the CFO's message or detecting the CEO's name in another context different from the chairman's highlights. Adding another indication improved the accuracy and the number of extracted gold standards. Once we confirm that we found the start of the chairman highlights, we continue extracting word by word until the end of the paragraph. This technique successfully extracted chairman highlights in most cases. However, some companies do not include chairman highlights in their communications. Algorithm 2 gives more details about the implemented script.

Extraction of other gold standard summaries: For the other gold standard summaries, we found common patterns in French financial reports where companies use common expressions to describe the financial highlights of the company activity. If the pattern occurs, we will extract line by line until the end of the paragraph. To process entries using these patterns, we used a rule-based algorithm 3

ALGORITHM 2: Chairman Highlights Extraction

Input: French Financial Report
Output: Chairman Message
Data: Corpus of financial reports from companies listed on CAC40 and CAC60

```
indications= ["Commentant", "déclaré", "indiqué"]
tokenizer = AutoTokenizer.from_pretrained("Jean-Baptiste/camembert-ner")
model =
  AutoModelForTokenClassification.from_pretrained("Jean-Baptiste/camembert-ner")
nlp = pipeline('ner', model=model, tokenizer=tokenizer,
  aggregation_strategy="simple")
for line in annual_report do
    read line                                      // read the report line by line
    ner = nlp(line)                        // apply the ner-cammembert on this line
    if chairman_higlights_condition exists in line and if entity_group=='PERSON'
      exists and PER["score"] ≥ 0.52 :  then
        Start_extraction = True                                  // Start extracting
        match += line                            // add line to the extracted text
    else if blank line and start_extracting == True then
        start_extracting = False                                  // stop extracting
    else if start_extracting == True:                         // continue extracting
     then
        match += line                                         // continue extracting
```

And finally, to extract the general highlights or the perspectives sections, we used the same algorithm as 3 but with different indications such as: other_indications =["communiqué de presse", "COMMUNIQUÉ DE PRESSE" , "Press release", "PRESS RELEASE", "Faits marquants", "FAITS MARQUANTS DU PREMIER SEMESTRE", "Faits marquants de l'année", "Principaux éléments du", "Message des présidents" , "Excellent semestre pour", "Excellente performance de", "Informations aux actionnaires", "Bonne dynamique de l'activité et des resultats", "Bon premier semestre", "Paris, le" "Excellent premier semestre pour", "Perspectives"]

5.4.4.6 Results and Challenges of the Labelling Process

The success rate for the chairman highlights was very promissing. The script worked 80% for Essilor (Sem and annual reports). For the other remaining reports, there is no chairman highlights. It means the rule-based scripts extracted all the chairman highlights once they are available. For AirLiquide, we got very good results with more than 80% success rate, especially for reports after 2010 since they follow a clear structure (Files before 2010 are not well structured). For Bouygues, the script extracted all the chairman highlights: they exist in around 40% of Semestrial and Annual reports. For other companies, we have Capgemini: 100% success rate, Sodexo: 100% success rate, Vinci 50% success rate Alstom 75% success rate, Arkema 95%

ALGORITHM 3: Extracting Financial Highlights

Input: French Financial Report
Output: Financial Highlights
Data: Corpus of financial reports from companies listed on CAC40 and CAC20

```
financial_highlights_indications = ["Le chiffre d'affaires du premier semestre","publie
  ses résultats", "Résultats annuels","Résultats du premier","PRESENTATION DE
  L'INFORMATION FINANCIERE","Presentation de l'information financiere",
  "Activité et informations financières" , "Rapport financier semestriel" "Résumé des
  résultats consolidés de l'année" , "Revue par segment de l'année 2013" , "Résultats
  du premier semestre" , "Perspectives confirmées pour", "Résultats semestriels",
  "RESULTATS DU DEUXIEME TRIMESTRE", "Deuxième trimestre", "CHIFFRE
  D'AFFAIRES OPERATIONNEL AU", "Chiffre d'affaires du premier"]
for line in annual_report do
    read line                                              // read the report line by line
    if financial_higlights_condition exist in line then
        Start_extraction = True                            // Start extracting
        match += line                                      // add line to the extracted text
        n_lines = n_lines+1                                // number of extracted lines
    else if blank_line and start_extracting == True and (n_lines > k = 2 ) then
        start_extracting = False                           // stop extracting
    else if start_extracting == True:                      // continue extracting
     then
        match += line                                      // continue extracting
        n_lines = n_lines+1                                // number of extracted lines
```

success rate, CreditAgricole 75% success rate, Dexia 90% success rate, Dassault 95% success rate, Klepierre 50% success rate.

The older reports (before 2008) are very challenging since they do not have a clear structure and come with a lot of noise caused by the pdf2text conversion. However, the new reports (from the last ten years) follow a clear structure which helps extract the summaries using heuristic rules. In addition, we have noticed that all the communications of listed companies in 2020 and 2021 included new parts describing the pandemic's impact on the company's activity. COVID-19 took an important part of companies' financial reports in the last two years. Therefore the reports of the last two years (2020 / 2021) were annotated manually because they have different structures, including long parts to describe the impact of the pandemic on the group's activity and how the company handled the new situation. Mainly the reports of the last two years include more narrative sections.

Verification of gold standard summaries: We performed a manual tuning of the dataset, deleting the non-relevant summaries. We ensured that we kept only meaningful summaries. We deleted the very short summaries (less than 20 words) and replaced them with more informative summaries. Finally, we manually added the gold standards not extracted by the algorithm. We deleted some outliers to ensure a normal distribution of lengths within the corpus. An outlier means a financial report longer than 250,000 tokens. In total, more than 50 % of the summaries needed manual correction. Therefore we can claim that our dataset is semi-manually annotated. It is evident that the heuristic rules will not generate a very well-annotated dataset. Otherwise, we do not need to train custom language models to summarise the report. Hence we needed to do post-editing and manual labelling in the cases where the algorithm failed to extract the good gold summary.

5.4.5 Dataset Exploration

5.4.5.1 Data Description

This summarisation corpus comprises approximately 1,703 financial narratives written in French, covering the period between 1995 and 2021. We used textstat[27] for counting tokens and sentences for all reports. The table 5.9 shows the statistics of the dataset by index (CAC40 and CAC20) details. We calculated the total number of tokens, the total number of sentences and the total number of reports by index.

Index	#Tokens	#Sentences	#Reports	#Gold summaries
CAC40	15,316,056	297,624	1,118	2,043
CAC20	7,529,046	142,486	591	947
Total	22,845,102	440,110	1,703	2,990

Table 5.9: Number (#) of tokens, sentences and reports relative to stock index

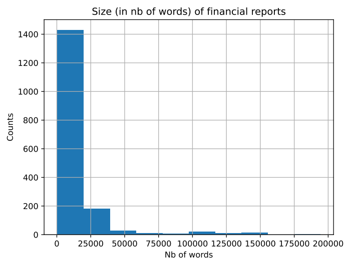
We divided the full text within annual reports into *training*, *validation*, and *testing* sets, providing both the full text of each annual report along with gold-standard summaries. The corpus is randomly split into training, validation and test sets using the ratio of 75%, 10%, 15%. For each set, we computed sentence and word statistics per report and per summary. They are reported in Table 5.10.

Data Type	Train	Val	Test	Total
# Report	1278	170	255	1703
# Gold summaries	2,255	296	439	2990
# words / report (average/median)	13576/5986	12809/6956	12979/5765	13410/6104
# sentences / report (average/median)	399/158	361/186	384/155	393/160
# words / summary (average/median)	248/206	255/213	238/198	247.6/205
# sentences / summary (average/median)	7.86/6	8.3/7	7.62/7	7.87/7
# compression ratio	0.06	0.06	0.07	0.06

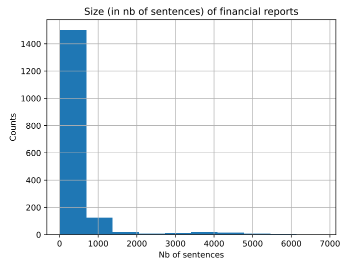
Table 5.10: French Financial Summarisation Dataset statistics (numbers are rounded)

[27] https://pypi.org/project/textstat/

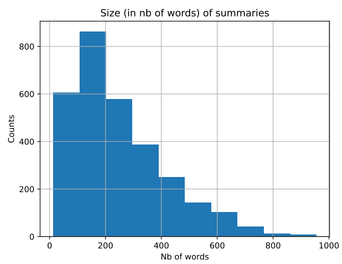
The Figures 5.14 5.15 5.16 5.17 show the distribution of the number of sentences and words in the financial reports and the respective summaries. Figure 5.18 shows the number of annual reports with n gold standard summaries (n could range from one to three).



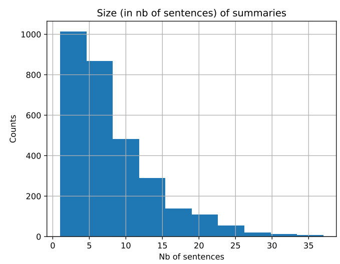
(a) Word Distribution for reports



(b) Sentence Distribution for reports

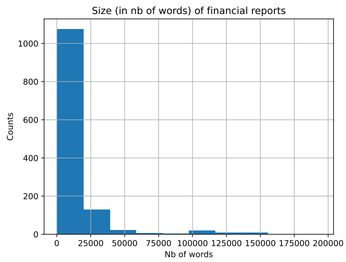


(c) Word Distribution for Summaries

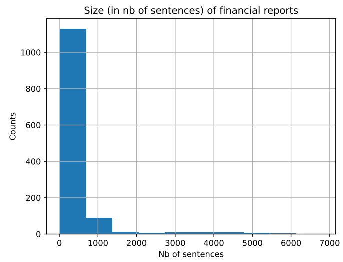


(d) Sentence distribution for summaries

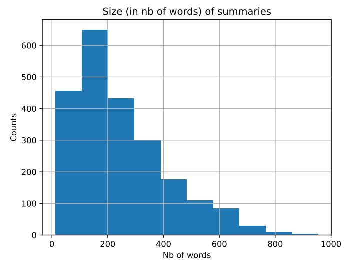
Figure 5.14: Distribution of word/sentence counts in the annual reports and their gold standard summaries (Plots on all the corpus)



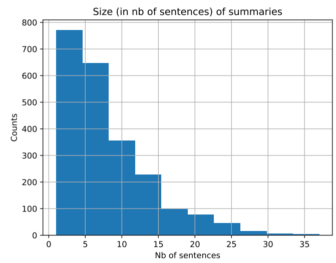
(a) Word Distribution for reports



(b) Sentence Distribution for reports

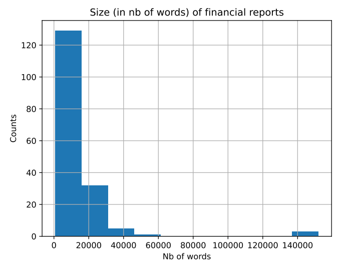


(c) Word Distribution for Summaries

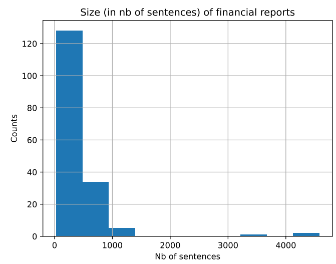


(d) Sentence distribution for summaries

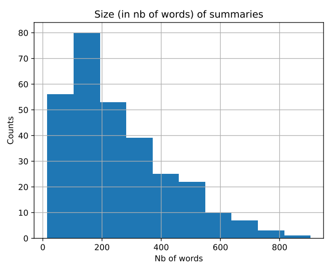
Figure 5.15: Distribution of word/sentence counts in the annual reports and their gold standard summaries (Training Split)



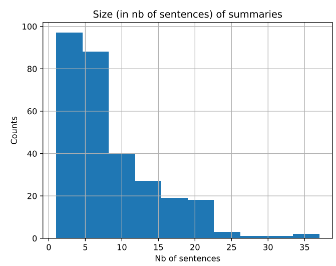
(a) Word Distribution for reports



(b) Sentence Distribution for reports

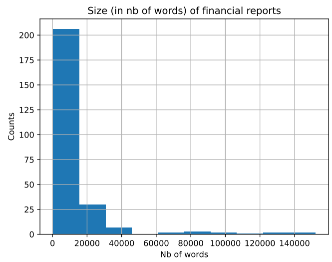


(c) Word Distribution for Summaries

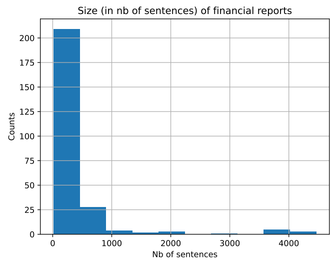


(d) Sentence distribution for summaries

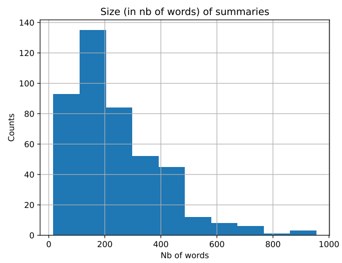
Figure 5.16: Distribution of word/sentence counts in the annual reports and their gold standard summaries (Validation Split)



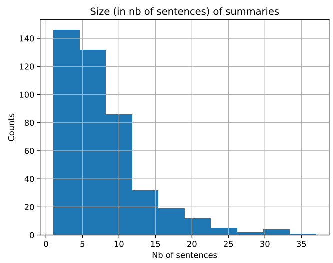
(a) Word Distribution for reports



(b) Sentence Distribution for reports

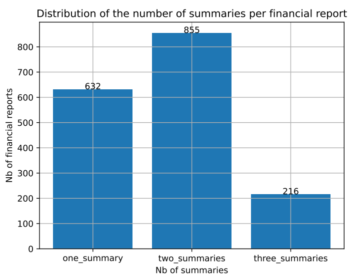


(c) Word Distribution for Summaries

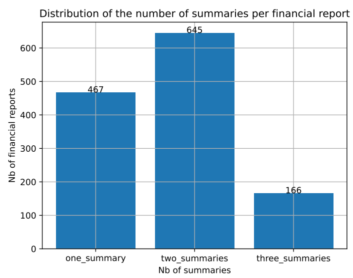


(d) Sentence distribution for summaries

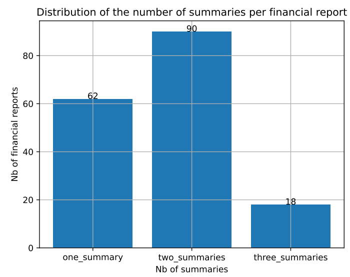
Figure 5.17: Distribution of word/sentence counts in the annual reports and their gold standard summaries (Test Split)



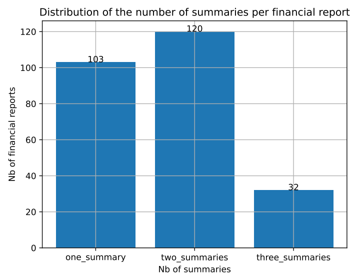
(a) All Corpus



(b) Train Split



(c) Validation Split



(d) Test Split

Figure 5.18: Distribution of number of gold standards per annual report (All/ Training / Validation/ Test)

5.4.5.2 Data Visualisation: Exploring the Dataset

To conclude, CoFiF Plus, the first French financial narrative summarisation dataset providing a comprehensive set of financial text written in French, was extracted from French financial reports published in PDF file format. It is composed of 1,703 reports from the most capitalised companies in France (Euronext Paris), covering a time frame from 1995 to 2021. The Figures 5.19a 5.19b 5.20 and 5.21 show the distribution of the unigrams, bigrams, treemap and word cloud respectively.

Figure 5.19a shows that the most common words used in our corpus are 'groupe', 'resultat', 'croissance' , 'rapport' and 'net'. Figure 5.19b shows that the most common words used in our corpus are 'resultat net', 'taux change', 'résultat opérationnel' , 'part groupe' and 'change constants'. Figure 5.20 plots the treemap of unigrams on the French dataset and finally figure 5.21 plots the french world cloud.

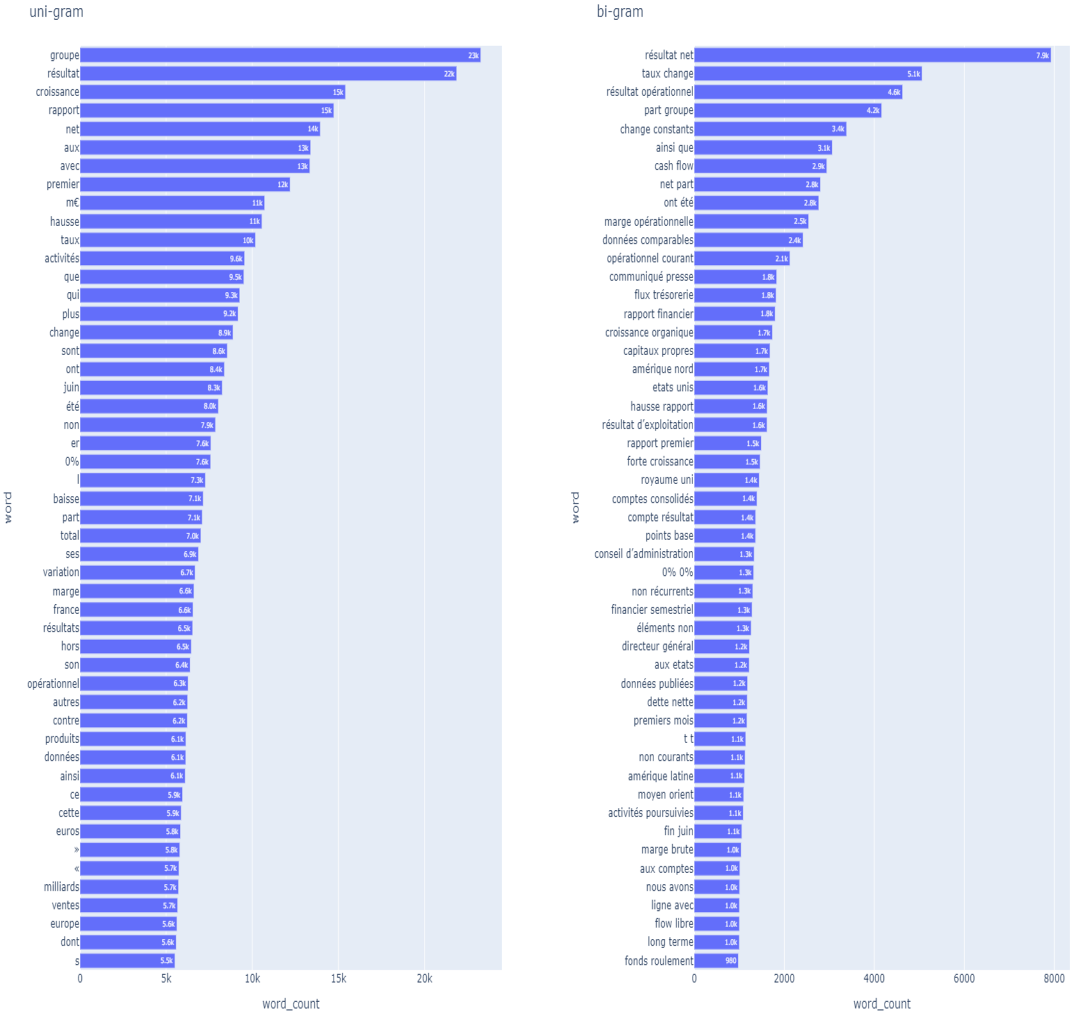
5.4.6 Dataset Scaling

The Cofif-plus dataset[28] used only French listed companies reports. Some examples of the extracted gold standard summaries are available in Appendix A.2.2.This dataset could be scaled to include Luxembourgish, Swiss, Belgian and some Canadian company reports in French. However, the different reporting frameworks tones and report lengths may create a bias in this dataset. That is why we focused on one financial market at this stage to find a homogeneous set of annual reports.

5.4.7 Summary

In this section, we presented a novel French financial narrative summarisation dataset composed of French annual, semester and trimestrial financial reports. Due to its careful company selection, we have demonstrated that it is a good representative corpus for finance communication in the French financial markets domain. This dataset will enhance the research in financial summarisation in the French language. This corpus is smaller than the English UK dataset that we previously described but comes with less noise and shorter summaries, which will encourage the use of the pretrained sequence2sequence French models such as Barthez and mBarthez and will help to explore another type of financial summarisation (medium size documents).

[28] `https://github.com/UCREL/CoFiF-Plus`



(a) unigrams distribution (b) bigrams distribution

Figure 5.19: unigrams and bigrams distribution for the French dataset

Figure 5.20: Tree map of French corpus

Figure 5.21: Word cloud of French Corpus

5.5 Ethics

Collecting a large financial annual report corpus for this research raises several critical ethical considerations. The financial report corpora were collected in full adherence to applicable legal requirements and regulations. The sources selected for data acquisition were reputable financial institutions and publicly available databases. The corpora were obtained from publicly available sources, and therefore, no explicit consent from individuals or organisations was required. However, it is essential to note that the extracted textual data will be used solely for research purposes within the context of this thesis.

Chapter 6

Long Document Financial Extractive Summarisation: Case Study on English Financial Annual Report Summarisation

There are more than 2,000 listed companies on the UK's London Stock Exchange, divided into 11 sectors. These companies are required to communicate their financial results at least twice in a single financial year. A financial annual report is a comprehensive document that provides detailed information about a company's financial performance and activities over the last fiscal year. UK annual reports are very lengthy documents, with around 80 pages on average. This creates a huge research interest in financial annual report summarisation. The creation of models that can scan and summarise financial documents could be a big research challenge, knowing that UK financial reports are very long and come into an unstructured format (free plain text). This encourages the use of natural language processing techniques to get the most from this textual data. As a proposed solution, we explore the use of different supervised and unsupervised automated methods.

In this chapter, we aim to benchmark a variety of summarisation methods on a set of different pre-trained transformers with different extraction techniques. The choice of benchmarking different pre-trained models is justified by the fact that each model has a different architecture of its encoder and decoder, and each model is pre-trained on a different task and corpus. In addition, we considered multiple evaluation metrics (embeddings-based and n-gram based) in order to investigate their differing behaviour and applicability on a dataset from the Financial Narrative Summarisation (FNS 2020/2021/2022) shared tasks, which is composed of annual reports published by firms listed on the London Stock Exchange and their corresponding summaries. We hypothesise that some evaluation metrics do not reflect true summarisation ability and propose a novel BRUGEscore metric as the harmonic mean of ROUGE-2 and BERTscore. We would also like to see if different variants of metrics behave differently on the same task. We performed a deep analysis of our summarisation results by plotting different correlation matrices. We also performed different boxplots and distribution plots in order to explore the distribution of our results. After that, we performed a statistical significance test on our results to verify whether they were statistically robust, and we finished our study with an adversarial analysis task with three different corruption methods in order to evaluate the evaluation robustness by testing how well the metrics handle these attacks.

6.1 Introduction

With the proliferation of firms worldwide, the volume of financial disclosures and financial texts (or narratives) in different languages and formats has risen dramatically, and therefore the study of natural language processing (NLP) methods that automatically summarise content has grown rapidly into a major research area (Zmandar et al., 2021e; Zmandar et al., 2021e; El-Haj et al., 2022d; El-Haj et al., 2020c).

In fact, financial reporting and communication requirements have been extended significantly in recent years, particularly following the 2008 financial crisis. Financial communications and investor relation management are becoming crucial parts of the financial markets and the fund management industry. Regulated financial markets mandate that all listed companies regularly communicate their financial activities to stakeholders by publishing financial reports and other financial narratives on a regular basis. Firms employ financial narratives to communicate with their stakeholders, including investors, shareholders, customers, employees, financial analysts, regulators, lenders, rating agencies, and suppliers. Through financial communications, stakeholders can assess how well the company is creating value.

In this chapter, we aim to create and evaluate summarisation benchmarks for UK financial narratives, and investigate the effect of long document methods and their interactions with various metrics, including ROUGE, in order to evaluate their suitability for this domain. Additionally, we will introduce a statistical testing method for system-generated financial summaries.

6.2 Evaluation Metrics Used in This Chapter

Summarising text is a complex task, and standard evaluation metrics such as accuracy, recall, and precision may not be suitable for text summarisation. In recent years, several metrics have been introduced that are specifically designed for evaluating the quality of machine-generated summaries. In this study, we used the following metrics: ngram-based metrics such as ROUGE (Lin, 2004) (coming with different variants), CIDEr (Vedantam et al., 2015), METEOR (Banerjee et al., 2005), and model-based metrics such as BERTScore (Zhang et al., 2020b) and Bleurt score (Sellam et al., 2020). All the metrics were previously presented and detailed in section 2.7.

For ROUGE (Lin, 2004), we used seven variations: ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-L, ROUGE-SU4, ROUGE-S1 and ROUGE-S2. We did not use an English stemmer, and we reported the mean of the F1 score of the different variants. The F1 score metric measures the harmonic mean of precision and recall for the generated summary. It is helpful in evaluating the overall quality of the summary and identifying false positives and false negatives.

Regarding BERTScore (Zhang et al., 2020b), we calculate two versions. The first named Bertscore1 which uses an encoder-only model {roberta-large-mnli}[1] (Liu et al., 2019b). The second is named Bertscore2 which uses an encoder-decoder model {Bart-large-mnli} from Facebook[2]. We report the mean of F1 scores in both cases. The motivation behind this is to test Bertscore with TED (Transformer Encoder-Decoder) (unidirectional) and TE (Transformer Encoder) architectures (bidirectional). The TED model (BART large) is pretrained with the aim of performing text generation, which makes it a good candidate for Bertscore whereas the TE model (Roberta) was trained using the Masked Language Modelling task. For Bleurt (Sellam et al., 2020), our implementation clones the original version[3] which is supported by Huggingface API. Also in this chapter, we will report **BRUGEscore** which is our novel proposed metric calculated as the harmonic mean of ROUGE-2 and BERTscore in order to combine elements of word overlap and embedding cosine similarity into one score (see equation 6.1).

$$BRUGEscore(BR) = \frac{2 \times \text{Rouge2} \times \text{Bertscore}}{\text{Rouge2} + \text{Bertscore}} \tag{6.1}$$

Table 6.1 describes in detail the features of the various used metrics, whether they are embeddings based or n-gram based.

6.3 UK Financial Narrative Summarisation

The employment of summarisation and natural language processing techniques, in general, has promising applications in the financial domain (El-Haj et al., 2019a). The first Financial Narrative Summarisation (FNS) shared task was held as part of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020) (El-Haj et al., 2020a; El-Haj, 2019). It involved the generation of structured summaries from financial narrative disclosures. In fact, the Financial Narrative Summarisation shared tasks (El-Haj et al., 2020c; Zmandar et al., 2021e; El-Haj et al., 2020a; El-Haj et al., 2022c) resulted in the first large-scale experimental results and state-of-the-art of financial report summarisation methods.

Prior works on UK annual report summarisation include (Orzhenovskii, 2021), which used a transformer-based encoder-decoder extractive summarisation approach based on the T5 pre-trained model.

[1] https://huggingface.co/roberta-large-mnli
[2] https://huggingface.co/facebook/bart-large-mnli
[3] https://github.com/google-research/bleurt

Metric	Embeddings?	LM	n-gram
ROUGE-1	No	N.A	1-gram
ROUGE-2	No	N.A	2-gram
ROUGE-3	No	N.A	3-gram
ROUGE-L	No	N.A	Sequence
ROUGE-SU4	No	N.A	2-gram
ROUGE-S1	No	N.A	2-gram
ROUGE-S2	No	N.A	2-gram
BERTScore1	Yes	roberta large	1-gram
BERTScore2	Yes	bart large	1-gram
BLEURT	Yes	BERT-lg	Sequence
Meteor	No	N.A	1-gram
CIDEr	No	N.A	1-gram
BRugeScore	Yes	N.A	2-gram

Table 6.1: Comparison of the used evaluation metrics features. LM: Language Model.

Singh (2020) proposed a Pointer Network and T5-based summarisation approach to extract then abstract relevant narrative sentences in a particular order to have a logical flow in the summary.

Li et al. (2020a) explored UK annual report summarisation using Determinantal Point Processes (DPPs) to build a statistical learning extractive financial narrative auto summariser.

Baldeon Suarez et al. (2020) combined financial word embeddings and knowledge-based features for financial narrative extractive text summarisation. They released their systems HULAT at FNS-2020.

La Quatra et al. (2020) developed an end-to-end training framework for financial report summarisation in English. The idea behind the system is to exploit the syntactic overlap between input sentences and ground-truth summaries to fine-tune pre-trained BERT language model to make it tailored to the specific financial context.

Arora et al. (2020) investigated different supervised and unsupervised ensemble method-based techniques which relatively perform better.

Vhatkar et al. (2020) formed a knowledge graph by considering words in the triplest: Subject(S), verb(V) and Object(O) ⟨S, V,O⟩. They modelled the summarisation task as a sentence classification and a triplet classification task which was performed using support vector regression and support vector machines.

Ait Azzi et al. (2020) have adopted an extractive technique which integrates neural networks and rule-based algorithms, with the expectation that it would be able to locate salient sentences or paragraphs from the data and pick out suitable candidates for creating final summaries.

Zheng et al. (2020) presented the SUMSUM financial summariser system, which was submitted to the Financial Narrative Summarisation Shared Task (FNS-2020). They developed a section-oriented extractive summarisation technique that is tailored for UK financial reports. Their system partitions the report's Table of Contents (ToC), divides the report into sections based on the ToC, and uses a BERT-based classifier for each section to decide if it should be included in the summary. So it is a way of turning the problem into a classification task.

Gokhan et al. (2021) created an unsupervised, extractive automatic financial summarisation system. The documents were first analysed, and a customised document set was produced containing the most important sections of the reports. Then, vector representations were generated for this intermediate dataset based on SentenceBERT. Afterwards, the vectors were clustered and sentences from each cluster were selected to form the final report summaries. In a nutshell, the author proposed a SentenceBERT-based clustering unsupervised approach and the results are still less competitive than the supervised, trained models.

Krimberg et al. (2021) presented a method for summarising financial documents using the Bag-of-Words (BOW) document representation and Term Frequency-Inverse Document Frequency (TFIDF) weights.

Litvak et al. (2021) describes a method for summarising financial texts that combines neural document modelling and genetic algorithms.

Khanna et al. (2022) applied the Longformer-Encoder-Decoder (LED) model in order to process lengthy financial reports. In addition, They claim that multi-stage fine-tuning helps the model in generalising better on specific domains (Eg. finance). They further examined the effect of the staged fine-tuning approach on the FNS 2022 dataset.

Shukla et al. (2022) proposed three summarisation systems for annual financial reports in English, Spanish and Greek languages. They worked on automatic methods for recognising key narrative sections and their use in creating financial report summaries. They have developed a novel approach to automatically identify the most important narrative sections of a report and their relative importance in the whole report (attributing them different weights). The most important sections are combined to form a system summary with a limit of 1000 words. They also demonstrate that their technique is able to adapt to various report formats and languages.

Vaca et al. (2022) presented three summarisation systems that were submitted to the financial narrative summarisation shared task (FNS-2022). They used a specific extractive summarisation technique for reports in English, which was based on a sequence classification task designed to identify the beginning of the summary. For Spanish and Greek datasets, since they were not extractive, they employed an abstractive strategy for each language by designing MariMari - a new Encoder-Decoder architecture in Spanish - as well as trained, multilingual Encoder-Decoder models for the Spanish systems. Regarding Greek, the summaries were obtained through a system that involved translating them to English, summarising them, and then translating the summaries back into Greek.

Foroutan et al. (2022) presented a multilingual automatic text summarisation system submitted for the financial narrative summarisation shared Task (FNS-2022). Two systems were designed to detect the commencement of a document's narrative part; the first is an abstractive summarisation model that has been customised to suit this task, while the second utilises extractive techniques based on trained span representations. The language models were fine-tuned on the three financial summarisation datasets in three languages (English, Spanish and Greek) used in the shared task. The proposed systems are very competitive with the sequence-to-sequence variant obtaining the highest ROUGE-2 F1 score on the test set for each of the languages.

El-Haj et al. (2022a) presented Hybrid TF-IDF and Clustering (HTAC), a methodology that utilises a clustering technique to create concise 1000-word summaries from lengthy annual reports on the FNS dataset. The Summariser utilises a hybrid extractive technique combining TF-IDF with K-Means clustering. It uses a statistical method to combine TF-IDF scores for each sentence of the input text with the Euclidean distance to the centre point of its cluster. Then, these scores are used to rank sentences, and the top-scoring sentences are incorporated into the summary up until it reaches 1000 words. Hence, this model removes unnecessary information and extracts the essential details into shorter and comprehensive summaries. The drawback of this methodology is that it generates a discontinued summary, which may sometimes be non-coherent.

Abdaljalil et al. (2021) investigated various methods by setting up two distinct challenges: a sentence-based and a section-based summarisation task. This one is adapted to the structure of the UK financial annual reports datsets. For the sentence-based summarisation, the problem was turned into a binary sentence classification problem, where sentences are assigned either 1 or 0 depending on whether they should be kept in the system summary or not. To create a labelled training set, they went through each sentence in the annual reports of the training dataset and compared them to their corresponding reference summaries. If it was present, it was given a label of 1; if not, it was labelled 0. They then used this annotated data to fine-tune the BERT model on this binary classification task. This model presents several drawbacks for the quantitative evaluation since it generates discontinued summaries, whereas the references are block-based summaries.

Vanetik et al. (2022) presented a method for summarising financial texts that incorporates multiple techniques for representing sentences and modelling documents using neural networks. The proposed approach is extractive in nature and follows the traditional process of ranking and selecting the most highly ranked sections of text (Vanetik et al., 2022). Vanetik et al. (2022) tested their system on the FNS dataset and their results surpassed the performance of the FNS topline. The method is also efficient in terms of time consumption alongside its impressive performance.

In comparison to prior works, we explore the impact of different transformer model architectures, the task, the preprocessing and data used to pre-train transformer models, as well as exploring correlations between some automated metrics within the task of summarising UK annual reports and the statistical testing of NLP models. Our work is distinct from previous general summarisation works as UK annual reports are long, unstructured in plain text, technically written, and subjective.

To address the memory efficiency issue of transformers, we cannot simply pass the entire input annual report and gold standard to the model and fine-tune it. Instead, we need to determine which parts of the report to pass to the transformer. Through dataset analysis, we found that the gold standards are typically extracted from the first third of the report, where the chairman or CEO message and financial highlights are usually located. Therefore, we will pass the first k tokens to the model, where k depends on the model architecture, pre-training, and memory efficiency. Then, the model will be trained to predict the first n tokens of the system summary. On the test dataset of 500 UK annual reports, the model will predict the first n tokens, and we will continue the extraction of the remaining k tokens by determining which part of the report matches the predicted n tokens (until we reach 1000 words). This approach transforms the summarisation problem into a task of predicting the start of the summary, allowing us to adapt sequence-to-sequence transformer models to summarise long documents where the reference summary is a continuously extracted part of the original text. We refer to this technique as the block-based summarisation approach. This technique surpasses the memory efficiency issue of some transformers and is motivated by the fact that reference summaries are extracted from the financial annual report as a block. To our best knowledge, this is the best approach for adapting encoder-decoder transformer models to summarise long documents.

We describe several techniques for summarisation in this chapter, including transformer-based (Khanna et al., 2022; Orzhenovskii, 2021), reinforcement learning-based as a suitable approach for maximising a predefined reward function (Topline) (Zmandar et al., 2021b), unsupervised learning using LSA (Gong et al., 2001), BERT extractive (Miller, 2019), and SBERT extractive summarisation (Reimers et al., 2019). We also compare the results of these techniques to four toplines and baseline summarisers, as we show later in the chapter, and finally, we use Lead-1000 (the first 1000 words) as a strong baseline summariser. We also compare the use of multi-reference versus using only one reference for preprocessing the training dataset.

6.4 Architecture Analysis

In this section, we will introduce the different transformer models used in our study. The most used techniques in recent years are based on the transformer encoder-decoder (TED) architecture introduced by Vaswani et al. (2017). In our experiments, we used various transformer models, including the T5 transformer (Raffel et al., 2020), LongFormer Encoder-Decoder (Beltagy et al., 2020), as well as BART, Pegasus, and BERT (Lewis et al., 2019; Zhang et al., 2020a; Devlin et al., 2019).

T5 transformer: (Raffel et al., 2020) is a text-to-text language model based on the Transformer architecture developed by Vaswani et al. (2017), where the input and output are always strings, pre-trained on multiple tasks (unsupervised and supervised). It is configured for 4096 maximum input tokens. T5 works well on a variety of tasks by prepending a different prefix to the input corresponding to each task (e.g., for translation: translate English to German; for summarisation:

summarise:).

LongFormer Encoder-Decoder (LED) (Beltagy et al., 2020) is a sequence-to-sequence language model which is based on BART's architecture and supports 'long document generative sequence-to-sequence tasks'. LED comes with two Pre-trained models: **allenai/led-base-16384** and **allenai/led-large-16384**. led-base-16384 was initialised from the BART-base since both models share the exact same architecture. Therefore, LED is suited to long document summarisation (maximum input of 16k tokens). However, its large version needs a high computational memory and requires a huge optimisation to run efficiently on a GPU cluster.

BART (Lewis et al., 2019) is a pre-trained model on the English language and is pre-trained by first corrupting text with a noising function and then learning a model to reconstruct the original text. It is an encoder-decoder (seq2seq) transformer model with a bidirectional encoder and an autoregressive decoder. That is why BART is a good candidate language model for text summarisation. However, one of BART's biggest limitations is that it accepts a maximum of 1,024 tokens, which is not very suitable for financial long narrative summarisation tasks. BART is available like all other models in the huggingface transformers open source library (Wolf et al., 2020).

Pegasus (Zhang et al., 2020a) is an encoder-decoder language model. The pre-training task of Pegasus is as follows: important sentences are removed or masked from an input document and are generated together as one output sequence from the remaining sentences. Pegasus achieves state-of-the-art summarisation performance on all 12 downstream tasks (measured by ROUGE and human evaluation) performed by Zhang et al. (2020a).

BERT (Devlin et al., 2019) (Bidirectional Encoder Representations from Transformers) is an encoder-only language model that was pre-trained using a masked language modelling objective and next sentence prediction on a large corpus comprising the 'Toronto Book Corpus' and 'Wikipedia'.

6.5 Summarisation Techniques

6.5.1 Transformer-based (Encoder Decoder) Summarisation

Memory is the bottleneck for long document summarisation (Saggion et al., 2003; Amini et al., 2006; Saggion et al., 2003; Amini et al., 2006; Tang et al., 2022b). Although efficient transformer models may reduce memory requirements of self-attention, the model still requires a lot of computational memory for the feed-forward layers. The most memory-efficient transformer is the Longformer encoder-decoder, which enables a maximum input of 16k tokens. Therefore, we cannot use transformers as a black box by simply passing the input annual report and the gold standard to the model and fine-tuning it. We need to consider which parts of the annual reports will be passed to the transformer. By analysing the dataset, we can prove that the gold standards are extracted from the first third of annual reports. Simply because the chairman's highlights or CEO's message or financial highlights are always mentioned at the beginning of financial narratives in order to capture the attention of shareholders and investors. After that, we will have the technical accounting and financial aspects of the report. Therefore, we will pass the first k tokens to our model. K depends on the model architecture, the pretraining of the model, and how memory-efficient the model is. Then, we will train the model to predict the first n tokens of the system summary. On the test dataset (500 UK annual reports), the model will predict the first n tokens (e.g., 128 tokens), and then we will continue the remaining k tokens by extraction after determining which part of the report matches the predicted n tokens. This method will transform our research problem from a summarisation problem to a task of predicting the start of the summary. It means our language model will predict the start of a good summary candidate. Then, we will continue the extraction ourselves. This is used to surpass the issue of memory efficiency of some transformers. In addition, this is motivated by the fact that the set of reference summaries is extracted from the financial annual report as a block. To the best of our knowledge, this is the best technique to adapt the sequence to sequence transformer models (encoder-decoder architecture) to summarise long documents where the reference summary is a continuously extracted part of the original text. We can name this technique as a block-based summarisation approach following (Orzhenovskii, 2021).

6.5.2 Using BERT and sBERT for sentence-level Summarisation

BERT Extractive is an unsupervised summarisation technique based on the method proposed by Miller (2019). This method works by first embedding the sentences, then running a clustering algorithm, finding the sentences that are closest to the cluster's centroids. For our benchmarking study, we used the {bert-large-uncased}[4] checkpoint. We set the maximum number of sentences for our summary to 30 sentences. This is equivalent to around 1,000 words. Sentence-based extractive summarisation transforms the task into a binary sentence classification task that assigns 1 to a given sentence if it is the closest to the cluster's centroids (to be kept in summary) and 0 if it is to be discarded. This technique would not be very efficient in our case due to the way the dataset was labelled.

SBERT Extractive is based on the project 'Sentence-BERT': Sentence Embeddings using Siamese BERT-Networks (Reimers et al., 2019). This project fine-tunes BERT / XLNet with a siamese or triplet network structure to produce semantically meaningful sentence embeddings that can be used in unsupervised scenarios such as summarisation via an embedding clustering algorithm. SBERT could be installed from the Pypi repository[5].

[4] `https://huggingface.co/bert-large-uncased`
[5] `https://pypi.org/project/sentence-transformers/0.2.0/`

6.5.3 Unsupervised Summarisation

LSA (Latent Semantic Analysis): LSA is an unsupervised algorithm used for dimensionality reduction. Gong et al. (2001) introduced the idea of using LSA in text summarisation, We used Latent Semantic Analysis (LSA) in order to generate summaries for our UK financial reports. LSA is able to identify semantically important sentences. The process starts with the creation of a term by sentence matrix A = $[A_1, A_2, \ldots, A_n]$ with each column vector A_i, representing the weighted term-frequency vector of sentence i in the document under consideration. If there are a total of m terms and n sentences in the document, then we will have an m $\times$ n matrix A for the document.

6.5.4 Baseline and Topline Summarisers

We compared the results of different systems to four topline and baseline summarisers—MUSE (Litvak et al., 2013a), POLY (Litvak et al., 2013b), TextRank (Mihalcea et al., 2004), and LexRank (Erkan et al., 2004b). In addition, we used Lead-1000 as a strong baseline summariser. LEAD-1000 is a very strong baseline for text summarisation. It selects the first 1,000 tokens of the input document. The LEAD method is known to be most effective for document summarisation of newspaper articles with a lower summarisation ratio because important sentences tend to appear in the first few sentences of a newspaper article (Narayan et al., 2018a). We tried to test it as a baseline for summarising long financial documents.

6.5.5 Reinforcement Learning Based Summarisation

In this section, we are presenting another topline in addition to MUSE that is based on reinforcement learning. The idea derives from the Pointer Networks and Copy Mechanism work. The proposed solution is denoted as **RNN-LSTM RL**, and it adopts a hybrid approach that combines extractive and abstractive techniques. It employs an actor-critic policy gradient with a rouge-2 reward function to connect the two components (extractor and abstractor).

Step 1: Train Custom Word2vec Financial Word Embeddings: This is motivated by the fact that the terminology used in financial disclosures diverges from everyday language. We used the English word embedding trained in chapter 4. Check our paper (Zmandar et al., 2021a) for more details on how to train a custom word2vec word embedding model for financial use cases.

Step 2: Preprocessing the Dataset for Training: For each summary sentence, we calculate its $ROUGE$ score with every sentence in the report and then select the sentence with the highest $ROUGE - 2$ value.

Step 3: Training an Extractor Agent: Using a hierarchical neural network model, the training involves comparing the sentences extracted by the pointer network with the proxy sentences obtained in the previous step as ground truths with the aim of minimising cross-entropy loss.

Step 4: Training an Abstractor Agent: The abstractor uses an LSTM (Long Short Term Memory layers) that receives the extracted content and generates paraphrased summary sentences. The abstractor network is designed as an approximation of the function that paraphrases an extracted document sentence into a concise summary sentence.

Step 5: Training a Reinforcement Learning Agent: The final comprehensive model is trained using a policy gradient algorithm using a basic actor-critic architecture. This process follows a similar approach as outlined in (Chen et al., 2018). The RL agent receives value estimation of input states and output actions in order to maximise a reward function. Hence, the agent learns by creating a loop of reward state-action.

Step 6: Decoding Approach on the test sample: To generate the final summary, we use a beam search decoding approach following the equation 6.2 where z^* is the generated summary. Beam search limits decoding to K potential hypotheses.

$$z^* = arg\ max\ p(z|x,\theta) = arg\ max\ \prod_{i}^{T_z} p(z_i|\{z_0, ..., z_{i-1}\}, x, \theta) \tag{6.2}$$

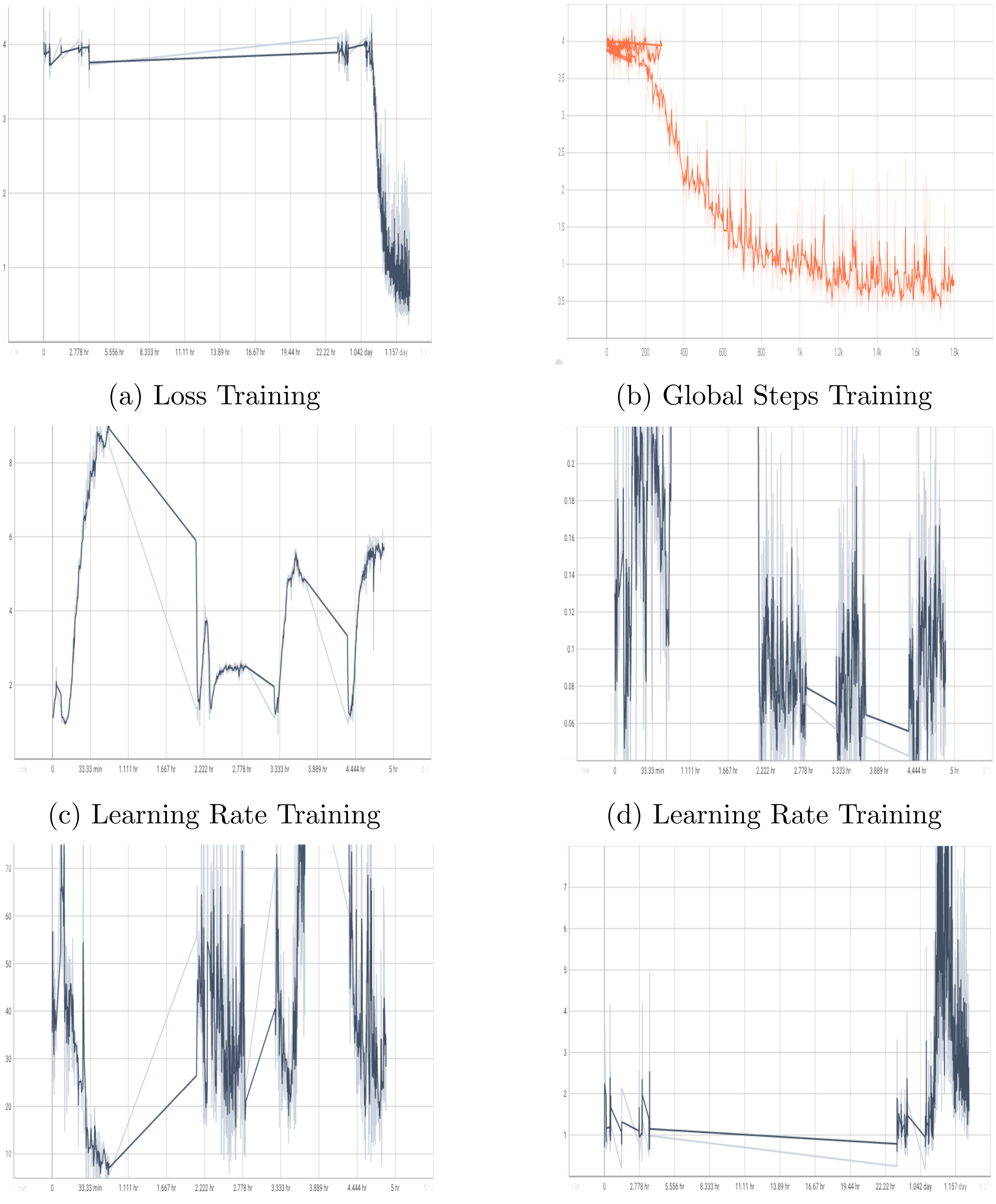
Experimental setup / Hardware used: In order to train our extractor, abstractor and RL models, we used a Tesla P100-PCIE GPU (based on NVIDIA's Turing architecture) with accelerated high RAM of 16 gigabytes on Google Colab Pro[6]. The GPU acceleration allowed us to train complex models and perform computations much faster, reducing training times and increasing overall efficiency.

Tensorboard Visualisation: We used the Google Tensorboard tool[7] to plot some statistics and monitoring charts from the training process on Google Colab. Tensorboard enables monitoring of the loss and the learning rate. We report the loss, lr and global steps in figure 6.1.

We should mention that reinforcement learning may sometimes generate random output or non-coherent output (Parnell et al., 2021; Narayan et al., 2018b) since it is trained to maximise a reward function (Rouge-2 in this case) rather than generate coherent text or focus on semantics. Furthermore, the output can not be fully reproducible since the RL agent will extract and then abstract. In this section, we will use the reinforcement learning summariser as a Topline system to be compared with transformers.

[6] `https://research.google.com/colaboratory/`

[7] `https://www.tensorflow.org/tensorboard`



(a) Loss Training

(b) Global Steps Training

(c) Learning Rate Training

(d) Learning Rate Training

(e) Learning Rate Training

(f) Learning Rate Training

Figure 6.1: Tensorboard Plotting of rnn-lstm-rl training

6.6 Experimental Methodology / Setup

6.6.1 Data Description

The dataset is composed of UK annual reports in English from the financial summarisation shared task (Zmandar et al., 2021e; El-Haj, 2019; El-Haj et al., 2020f; El-Haj et al., 2021c). The dataset around 3,800 annual reports for firms listed on the London Stock Exchange (LSE) covering the period between 2002 and 2017. The average length of an annual report is 52,000 tokens. The dataset includes more than 9,800 gold standard summaries. The dataset is randomly split into training (75%), testing and validation (25%). The dataset was presented and described in section 5.2.

6.6.2 Preparing the Dataset for Training

We are using multiple references for each annual report (between three and seven) to make the process more objective since we do not have a human-generated reference summary as a good gold standard. When preparing the dataset, the gold standard summaries were extracted from the annual reports, and we used several references to make the process more objective. The gold standards used are Financial Highlights, Letter to the Shareholders, Financial Statements and Auditor's Report. In order to prepare the training and validation dataset pairs, we need to choose one gold standard summary per annual report. Our selection criteria will be based on the aim of maximising the ROUGE-2 metric (Zmandar et al., 2021e). Hence, we will set our reward function as ROUGE-2 and select the gold standard summary that maximises ROUGE-2 with the annual report. The data could be served in JSON or CSV format with two columns representing, respectively the input document and the summary.

6.6.3 Research Methodology

The main research question is about the gold summary standard and whether we should opt for one or several summaries. We investigated whether using multiple gold standard summaries would improve the performance of summarisation models. We trained T5, Pegasus and BART using two different strategies: The first strategy involved using all the available gold standards. This means we will create several pairs for every report. This is an objective way of preparing the dataset. We will not decide which gold standard is better. The second strategy was to choose only one gold summary, which maximises the ROUGE metric as described in section 6.6.2, which was the aim of the FNS task. The result of our preliminary study showed that training on a multi-referenced dataset did not significantly improve the ROUGE result. In addition, it consumed much more computational power and time. Therefore, we chose to train our models using only one reference summary per annual report. We set our reward function as ROUGE-2 and selected the gold standard summary that maximised the ROUGE-2 score with the annual report. This enables our system sumamrisers to maximise the Rouge metric with all the reference summaries.

6.6.4 Hyperparameter Search

One of the common ways to select the fine-tuning parameters is using a comprehensive grid search that was used by Zhang et al. (2020a). However, due to the significant computational power and time required, we opted for a simpler strategy. We selected hyperparameters that maximise our models' input and target lengths, as detailed in Table 6.2. (see appendix B.1 for more detailed hyperparameters) For example, for T5, we have two important hyperparameters: max input length and max output length. The first was fixed to 4,000 tokens, and the second to 512 tokens on our University's High-End Computing (HEC) infrastructure. Then, on the testing dataset, we pushed the limits to the maximum of the GPU, and we were able to generate 512 tokens; then, we continued the remaining by extraction.

Transformer	Parameters				
	model name	max input	max target	batch size	train epochs
T5	t5-small	4096	512	4	5
LED base	allenai/led-base-16384	8000	1000	4	5
LED large	allenai/led-large-16384	4096	512	4	5
Pegasus	google/pegasus-large	1024	256	4	5
BART	facebook/bart-base	1024	128	4	5

Table 6.2: description of hyperparameters during training on the FNS dataset

6.6.5 Training

In order to pre-process the input dataset, we need to convert the text into numbers before passing it to the transformer model. For this, we use the AutoTokeniser from the HuggingFace hub. Then, the tokens are passed to the HuggingFace Trainer API, which provides a Trainer class to help in finetuning any pre-trained language model. We then prepare the environment to run the Trainer API, which should be a GPU or TPU. The trainer API will take several minutes to fine-tune the model for one epoch. We set the Trainer to evaluate at the end of each epoch. We need to set up an optimiser and a learning rate scheduler. The optimiser used by the Trainer is AdamW. The learning rate used by the Trainer is 1e-5. The training was done on a Conda[8] environment 6.1 on the High-End Computing Cluster of Lancaster University [9].

```
channels:
  - pytorch
  - defaults
  - conda-forge
dependencies:
  - python=3.8
```

[8]https://docs.conda.io/en/latest/

[9]https://www.lancaster.ac.uk/iss/info/IThandouts/hec/HEC-flyer.pdf

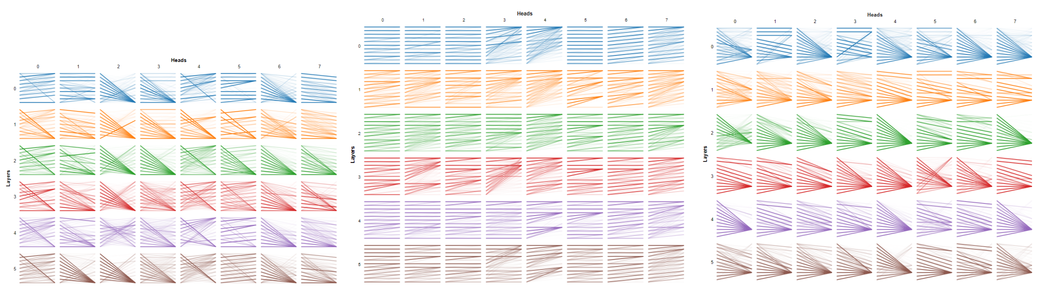
```
  - pip
  - pytorch=1.7.0
  - cudatoolkit=10.2
  - conda-forge::git
  - conda-forge::git-lfs
```

Listing 6.1: Conda Environment fine-tuning task

Training Hardware The training was performed on the short queue of a High-End Computing cluster using one GPU and 4 CPUs. The maximum run time was set to 12 hours, and we used a 40 GB memory. The training was performed using a Conda PyTorch virtual environment. On average, every transformer will need between two and four hours for training. A connection was created between the HEC and the wandb platform[10] in order to monitor the training of the transformer.

6.6.6 T5 Small Model Visualisation

Figure 6.2a, Figure 6.2b and Figure 6.2c show an internal visualisation of attention across all of the model's layers and heads for a particular input ("UBS agrees to buy Credit Suisse") and output ("UBS will buy Credit Suisse for two billion dollars ") of the fine-tuned T5 small model. The three figures represent respectively, the encoder, decoder and a cross representation(Encoder-Decoder). The attention heads are organised in the format of tables, where columns show heads and rows indicate layers. The model view (illustrated in Figure 6.2) lets us browse the attention heads across all layers quickly and see how attention patterns evolve throughout the model. This view is particular to the specific input and output mentioned above.



(a) T5 small: The model view of the fine-tuned T5 small language model encoder (b) T5 small: The model view of the fine-tuned T5 small language model decoder (c) T5 small: A cross-model view of the fine-tuned T5 small language model

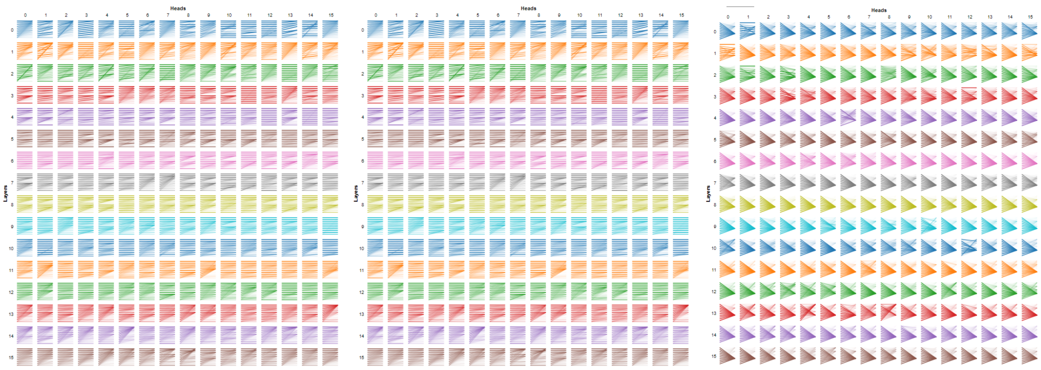
Figure 6.2: T5 small: the model view of the encoder and decoder of the finetuned language model

6.6.7 Pegasus Model Visualisation

Figure 6.3a, Figure 6.3b and Figure 6.3c show an internal visualisation of attention across all of the model's layers and heads for a particular input ("UBS agrees to buy

[10]Wandbai: `https://wandb.ai/`

Credit Suisse") and output ("UBS will buy Credit Suisse for two billion dollars") of the fine-tuned Pegasus model. The three figures represent respectively, the encoder, decoder and a cross representation(Encoder-Decoder). The attention heads are organised in the format of tables, where columns show heads and rows indicate layers. The model view (illustrated in Figure 6.3) lets us browse the attention heads across all layers quickly and see how attention patterns evolve throughout the model. This view is particular to the specific input and output mentioned above.



(a) Pegasus: The model view of the fine-tuned T5 small language model encoder (b) Pegasus: The model view of the fine-tuned T5 small language model decoder (c) Pegasus: A cross-model view of the fine-tuned T5 small language model

Figure 6.3: Pegasus: the model view of the encoder and decoder of the finetuned language model

6.7 Results and Analysis

In this section, we start by reporting the performance of the summarisation models using ROUGE and other evaluation metrics presented in section 6.2. Then, we will analyse these results and finish by performing a data visualisation analysis of the results. We will report on the distribution of the results and how much the different metrics are correlated with each other.

6.7.1 Results of Financial Narrative Summarisation

In this study, we used metrics that support multiple references to evaluate the performance of our models. To compute the Rouge scores between the system summary and all the gold standards, we used the Rouge.2.0 java jar[11] file for ROUGE evaluation and we removed English stop-words. For other metrics, we used the implementation from the original authors or the implementation of the Hugging Face team on the datasets library[12]. For all the metrics, we compute the score between the system summary and all the gold standards. Average F1 scores were reported

[11] https://github.com/kavgan/ROUGE-2.0

[12] https://huggingface.co/docs/datasets/index

for each metric, including different variants of the Rouge score (R1, R2, R3, R-L, R-SU4, R-S1, R-s2), BERT score, Meteor, Cider and Bleurt scores. To compute the embedded representation, we used, as stated previously, the 'Roberta-large-mnli' and 'Bart-large-mnli' language models for BERTScore1 and BERTScore2, respectively. Then, for evaluation, we opt for evaluation against all the gold standards (up to seven per annual report). We affirm that it penalises our system summaries that are apparently or qualitatively good; however, this is the most objective way to evaluate our summary, at least with such an extractive dataset. The idea behind creating several gold standard summaries was motivated by the fact that we do not know what exactly a good financial report gold summary is.

Table 6.3 reports ROUGE-1, ROUGE-2 and ROUGE-3 Recall, Precision and F-measure scores with standard deviations measured on the FNS test dataset. ROUGE-2 F1 measure is used for ranking the results. Table 6.4 reports ROUGE-L and ROUGE-SU4 Recall, Precision and F-measure scores with the standard deviations measured on the FNS test dataset. ROUGE-L F1 measure was used to rank the results. Table 6.5 represents ROUGE-S1 and ROUGE-S2 Recall, Precision and F-measure scores with the standard deviations measured on the FNS test dataset. ROUGE-S1 F1 measure was used to rank the results. Table 6.6 reports two bertscores Recall, Precision and F-measure scores with the standard deviations. BERTscore1 was calculated using (roberta-large-mnli), and BERTscore2 uses (bart-large-mnli) to calculate the embeddings. Table 6.7 reports Bleurt scores measured on the FNS test dataset using 'bleurt-large-512' checkpoint. And finally, table 6.8 reports meteor, cider and bruge scores.

For the naming of models, we use the convention ⟨transformer⟩-⟨version⟩-⟨tokens⟩ where ⟨transformer⟩ is the name of the transformer, ⟨version⟩ is base, small or large and ⟨tokens⟩ is the number of maximum target tokens. In other terms, T5-small-96 means that we generate the first 96 tokens using a fine-tuned version of T5 small, and we extract the remaining words to reach the 1000-word limit.

System / Metric	**R-1/R**	**R-1/P**	**R-1/F**	**R-2/R**	**R-2/P**	**R-2/F**	**R-3/R**	**R-3/P**	**R-3/F**
T5-Small-96	0.587	0.451	0.496	0.472	0.326	0.376	0.444	0.293	0.342
	0.14	0.1	0.09	0.16	0.1	0.11	0.17	0.1	0.11
LED-Base-128	0.578	0.45	0.492	0.461	0.327	0.37	0.431	0.294	0.337
	0.14	0.11	0.09	0.17	0.1	0.11	0.18	0.1	0.12
LED-Base-256	0.578	0.45	0.492	0.46	0.326	0.369	0.43	0.293	0.336
	0.14	0.11	0.09	0.17	0.1	0.11	0.18	0.1	0.12
LED-Base-1000	0.577	0.449	0.491	0.459	0.325	0.368	0.43	0.292	0.335
	0.14	0.11	0.09	0.17	0.1	0.11	0.18	0.1	0.12
PEGASUS	0.583	0.421	0.476	0.463	0.294	0.35	0.433	0.259	0.316
	0.14	0.1	0.09	0.17	0.1	0.12	0.18	0.1	0.12
T5-multi-references	0.535	0.437	0.467	0.402	0.306	0.335	0.368	0.273	0.3
	0.14	0.1	0.09	0.17	0.11	0.11	0.18	0.11	0.12
T5-Small-256	0.551	0.421	0.463	0.421	0.293	0.332	0.389	0.26	0.297
	0.14	0.11	0.09	0.17	0.11	0.12	0.18	0.11	0.12
Pegasus-multi-references	0.571	0.405	0.462	0.446	0.276	0.331	0.415	0.241	0.296
	0.14	0.1	0.09	0.18	0.1	0.11	0.19	0.1	0.12
T5-Small-512	0.523	0.439	0.463	0.385	0.306	0.328	0.351	0.273	0.293
	0.14	0.11	0.09	0.16	0.11	0.11	0.17	0.11	0.12
BART	0.519	0.427	0.453	0.381	0.295	0.317	0.348	0.264	0.284
	0.15	0.1	0.09	0.18	0.11	0.12	0.19	0.11	0.12
BART-multi-references	0.531	0.414	0.45	0.394	0.281	0.313	0.359	0.247	0.277
	0.14	0.1	0.09	0.17	0.1	0.11	0.18	0.11	0.12
LED-LARGE	0.558	0.386	0.444	0.427	0.252	0.308	0.395	0.218	0.271
	0.14	0.09	0.08	0.17	0.09	0.11	0.18	0.09	0.11
LEAD-1000	0.558	0.386	0.443	0.427	0.252	0.307	0.395	0.217	0.271
	0.14	0.09	0.08	0.17	0.09	0.11	0.18	0.09	0.11
RNN-LSTM-RL	0.516	0.436	0.459	0.38	0.22	0.27	0.336	0.132	0.183
	0.13	0.09	0.08	0.14	0.07	0.08	0.14	0.06	0.07
MUSE	0.483	0.413	0.433	0.311	0.198	0.234	0.239	0.112	0.147
	0.11	0.08	0.06	0.11	0.06	0.07	0.1	0.05	0.05
LSA	0.402	0.285	0.321	0.192	0.119	0.14	0.14	0.077	0.094
	0.1	0.08	0.06	0.1	0.05	0.06	0.09	0.05	0.05
SBERT-extractive	0.41	0.278	0.322	0.197	0.115	0.139	0.144	0.072	0.091
	0.07	0.07	0.05	0.07	0.05	0.04	0.07	0.04	0.04
BERT-extractive	0.343	0.304	0.312	0.163	0.123	0.134	0.119	0.077	0.088
	0.08	0.07	0.05	0.07	0.05	0.05	0.07	0.04	0.04
LEXRANK	0.337	0.269	0.264	0.193	0.107	0.12	0.143	0.054	0.067
	0.13	0.13	0.09	0.11	0.08	0.07	0.11	0.05	0.05
POLY	0.324	0.253	0.274	0.147	0.088	0.105	0.086	0.039	0.05
	0.08	0.07	0.05	0.07	0.04	0.04	0.06	0.03	0.03
TEXTRANK	0.414	0.118	0.172	0.229	0.044	0.07	0.165	0.021	0.036
	0.11	0.07	0.08	0.11	0.04	0.05	0.12	0.02	0.03

Table 6.3: ROUGE-1 and ROUGE-2 and ROUGE-3 Recall, Precision and F-measure scores with standard deviations measured on the FNS test dataset. ROUGE-2 F1 measure is used for ranking the results.

System / Metric	R-L / R	R-L / P	R-L / F	R-SU4 / R	R-SU4 / P	R-SU4 / F
T5-Small-96	0.559 0.14	0.449 0.1	0.487 0.1	0.515 0.15	0.369 0.1	0.417 0.1
LED-BASE-256	0.552 0.14	0.447 0.1	0.484 0.1	0.503 0.16	0.369 0.1	0.412 0.1
LED-BASE-128	0.553 0.14	0.447 0.1	0.484 0.1	0.504 0.16	0.37 0.1	0.413 0.1
LED-BASE-1000	0.55 0.14	0.446 0.1	0.482 0.1	0.502 0.16	0.368 0.1	0.411 0.1
PEGASUS	0.547 0.15	0.423 0.1	0.467 0.1	0.507 0.16	0.338 0.1	0.394 0.11
T5-multi-references	0.511 0.14	0.425 0.1	0.454 0.1	0.449 0.16	0.351 0.1	0.381 0.1
PEGASUS-multi-references	0.531 0.15	0.408 0.1	0.451 0.1	0.492 0.16	0.321 0.1	0.377 0.11
T5-Small-256	0.52 0.14	0.416 0.1	0.451 0.1	0.468 0.16	0.338 0.1	0.378 0.11
T5-Small-512	0.503 0.13	0.423 0.1	0.45 0.1	0.434 0.15	0.353 0.1	0.375 0.1
BART	0.496 0.14	0.414 0.1	0.44 0.1	0.431 0.17	0.342 0.11	0.365 0.11
BART-multi-references	0.502 0.14	0.404 0.1	0.437 0.1	0.442 0.16	0.328 0.1	0.361 0.1
LEAD-1000	0.514 0.15	0.387 0.09	0.431 0.09	0.474 0.16	0.3 0.09	0.356 0.1
LED-LARGE	0.515 0.15	0.387 0.09	0.431 0.09	0.475 0.16	0.3 0.09	0.356 0.1
RNN-LSTM-RL	0.493 0.13	0.398 0.09	0.431 0.09	0.437 0.13	0.202 0.06	0.268 0.07
MUSE	0.486 0.11	0.381 0.07	0.419 0.07	0.375 0.11	0.201 0.06	0.253 0.05
LSA	0.382 0.08	0.239 0.06	0.287 0.05	0.266 0.1	0.155 0.06	0.187 0.06
SBERT-extractive	0.371 0.06	0.228 0.05	0.276 0.04	0.275 0.07	0.15 0.05	0.187 0.04
BERT-extractive	0.309 0.06	0.239 0.05	0.263 0.04	0.23 0.07	0.161 0.05	0.182 0.05
LEXRANK	0.249 0.12	0.298 0.1	0.253 0.08	0.253 0.12	0.117 0.08	0.14 0.07
TEXTRANK	0.282 0.11	0.226 0.07	0.242 0.07	0.302 0.11	0.048 0.04	0.079 0.05
POLY	0.27 0.08	0.182 0.05	0.212 0.05	0.213 0.07	0.105 0.04	0.135 0.04

Table 6.4: ROUGE-L and ROUGE-SU4 Recall, Precision and F-measure scores with the standard deviations measured on the FNS test dataset. ROUGE-L F1 measure is used to rank the results

System / Metric	R-S1 / R	R-S1 / P	R-S1 / F	R-S2 / R	R-S2 / P	R-S2 / F
T5-Small-96	0.472	0.326	0.374	0.469	0.323	0.371
	0.16	0.1	0.11	0.16	0.1	0.11
LED-Base-128	0.461	0.327	0.37	0.458	0.323	0.366
	0.17	0.1	0.11	0.17	0.1	0.11
LED-Base-256	0.46	0.326	0.369	0.457	0.322	0.365
	0.17	0.1	0.11	0.17	0.1	0.11
LED-Base-1000	0.459	0.325	0.368	0.456	0.321	0.364
	0.17	0.1	0.11	0.17	0.1	0.11
PEGASUS	0.463	0.294	0.35	0.459	0.29	0.346
	0.17	0.1	0.12	0.18	0.1	0.12
T5-multiple-references	0.402	0.306	0.335	0.398	0.302	0.331
	0.17	0.11	0.11	0.17	0.11	0.11
T5-Small-256	0.421	0.293	0.332	0.418	0.289	0.328
	0.17	0.11	0.12	0.17	0.11	0.12
PEGASUS-multiple-references	0.446	0.276	0.331	0.443	0.272	0.327
	0.18	0.1	0.11	0.18	0.1	0.12
T5-Small-512	0.385	0.306	0.328	0.381	0.302	0.324
	0.16	0.11	0.11	0.16	0.11	0.11
BART	0.381	0.295	0.317	0.378	0.292	0.314
	0.18	0.11	0.12	0.18	0.11	0.12
BART-multiple-references	0.394	0.281	0.313	0.39	0.277	0.309
	0.17	0.1	0.11	0.17	0.11	0.11
LED-Large	0.427	0.252	0.308	0.424	0.249	0.304
	0.17	0.09	0.11	0.17	0.09	0.11
LEAD-1000	0.427	0.252	0.307	0.423	0.249	0.303
	0.17	0.09	0.11	0.17	0.09	0.11
RNN-LSTM-RL	0.38	0.22	0.27	0.38	0.188	0.244
	0.14	0.07	0.08	0.14	0.07	0.08
MUSE	0.311	0.198	0.234	0.307	0.173	0.214
	0.11	0.06	0.07	0.11	0.06	0.06
LSA	0.192	0.119	0.14	0.185	0.11	0.131
	0.1	0.05	0.06	0.1	0.05	0.06
SBERT-EXTRACTIVE	0.197	0.115	0.139	0.193	0.106	0.131
	0.07	0.05	0.04	0.07	0.04	0.04
BERT-EXTRACTIVE	0.163	0.123	0.134	0.16	0.114	0.127
	0.07	0.05	0.05	0.07	0.05	0.05
LEXRANK	0.193	0.107	0.12	0.191	0.093	0.109
	0.11	0.08	0.07	0.11	0.07	0.07
POLY	0.147	0.088	0.105	0.142	0.076	0.094
	0.07	0.04	0.04	0.07	0.04	0.04
TEXTRANK	0.229	0.044	0.07	0.225	0.038	0.061
	0.11	0.04	0.05	0.11	0.03	0.04

Table 6.5: ROUGE-S1 and ROUGE-S2 Recall, Precision and F-measure scores with the standard deviations measured on the FNS test dataset. ROUGE-S1 F1 measure is used to rank the results

System / Metric	BE1 / R	BE1 / P	BE1 / F	BE2 / R	BE2 / P	BE2 / F
T5-Small-96	0.919 0.09	0.903 0.10	0.910 0.10	0.874 0.12	0.804 0.12	0.830 0.12
LED-base-128	0.910 0.10	0.891 0.10	0.899 0.10	0.860 0.13	0.792 0.12	0.816 0.12
LED-base-256	0.907 0.10	0.889 0.10	0.896 0.10	0.856 0.13	0.789 0.12	0.813 0.12
LED-base-1000	0.904 0.10	0.886 0.10	0.894 0.10	0.853 0.13	0.786 0.11	0.810 0.11
BART	0.859 0.10	0.847 0.10	0.852 0.10	0.802 0.14	0.757 0.13	0.774 0.13
Pegasus	0.858 0.10	0.840 0.10	0.847 0.10	0.804 0.13	0.734 0.10	0.759 0.11
T5-multiple-references	0.854 0.09	0.843 0.09	0.847 0.09	0.798 0.12	0.756 0.11	0.772 0.11
T5-Small-256	0.845 0.10	0.830 0.10	0.837 0.10	0.790 0.12	0.732 0.10	0.753 0.11
T5-Small-512	0.840 0.09	0.828 0.09	0.833 0.09	0.782 0.11	0.746 0.11	0.759 0.11
Pegasus-multiple-references	0.824 0.10	0.805 0.09	0.813 0.09	0.774 0.12	0.707 0.09	0.730 0.10
BART-multiple-references	0.809 0.07	0.794 0.07	0.800 0.07	0.753 0.10	0.706 0.09	0.723 0.09
LSA	0.787 0.03	0.778 0.03	0.782 0.03	0.668 0.05	0.640 0.03	0.651 0.04
SBERT-extractive	0.790 0.03	0.775 0.02	0.781 0.02	0.665 0.04	0.638 0.03	0.647 0.02
Lead-1000	0.788 0.07	0.765 0.06	0.774 0.06	0.737 0.11	0.671 0.07	0.694 0.08
LED-large	0.788 0.07	0.765 0.06	0.774 0.06	0.737 0.11	0.671 0.07	0.694 0.08
BERT-extractive	0.779 0.03	0.766 0.02	0.771 0.02	0.651 0.04	0.623 0.02	0.632 0.02
RNN-LSTM-RL	0.763 0.05	0.763 0.04	0.761 0.04	0.653 0.06	0.649 0.06	0.647 0.06
MUSE-TOPLINE	0.762 0.04	0.754 0.04	0.756 0.04	0.673 0.05	0.648 0.05	0.655 0.05
LexRank	0.739 0.05	0.728 0.05	0.732 0.05	0.592 0.09	0.579 0.06	0.580 0.07
TextRank	0.735 0.06	0.721 0.05	0.727 0.05	0.591 0.09	0.572 0.06	0.576 0.07
POLY-BASELINE	0.726 0.03	0.725 0.02	0.723 0.02	0.583 0.04	0.557 0.03	0.565 0.03

Table 6.6: BE1 and BE2 Recall, Precision and F-measure scores with the standard deviations. BE1 stands for BERT score (roberta-large-mnli) and BE2 stands for BERT score (bart-large-mnli). BE1 F1 measure is used for ranking the results.

System / Metric	**bleurt**	**std bleurt**
T5-Small-96	-0.836972	0.32
LED-base-128	-0.849750	0.33
LED-base-256	-0.856580	0.33
LED-base-1000	-0.866264	0.33
Pegasus	-0.925372	0.29
BART	-0.928474	0.26
LSA	-0.945594	0.19
SBERT-extractive	-0.973918	0.20
T5-Small-256	-0.976840	0.26
T5-multiple-references	-0.977308	0.24
Pegasus-multi-references	-0.980006	0.26
BERT-extractive	-0.987254	0.19
T5-Small-512	-1.000086	0.21
RNN-LSTM-RL	-1.027724	0.19
Lead-1000	-1.039358	0.19
LED-large	-1.040436	0.19
BART-multi-references	-1.040898	0.20
MUSE-TOPLINE	-1.045138	0.16
LexRank	-1.051438	0.17
POLY-BASELINE	-1.060618	0.16
TextRank	-1.074088	0.16

Table 6.7: Bleurt scores measured on the FNS test dataset. We used 'bleurt-large-512' model.

System / Metric	meteor	cider	Bruge
T5-Small-96	0.184	0.009571	0.532
LED-base-256	0.182	0.011177	0.523
LED-base-128	0.182	0.011163	0.524
LED-base-1000	0.182	0.011044	0.521
T5-multiple-references	0.179	0.010778	0.480
T5-Small-512	0.179	0.008210	0.471
BART	0.176	0.007791	0.462
RNN-LSTM-RL	0.175	0.012803	0.399
Pegasus	0.174	0.007684	0.495
T5-Small-256	0.173	0.005561	0.475
BART-multi-references	0.171	0.005643	0.450
Pegasus-multi-references	0.169	0.007101	0.470
MUSE-TOPLINE	0.163	0.013155	0.357
Lead-1000	0.162	0.004817	0.440
LED-large	0.162	0.004817	0.441
LSA	0.160	0.005080	0.237
SBERT-extractive	0.159	0.004418	0.236
TextRank	0.144	0.001178	0.128
BERT-extractive	0.121	0.006085	0.228
POLY-BASELINE	0.109	0.003303	0.183
LexRank	0.088	0.006431	0.206

Table 6.8: METEOR, Cider and Bruge scores on the FNS dataset

6.7.2 Analysis of Results

The results shown in different tables suggest that T5 is the best text-to-text model for this dataset, performing well alongside Longformer Encoder-Decoder. The LED base is memory-efficient and performs very well on the dataset, while LED Large did not perform as well due to limited GPU memory (unability to increase the maximum input length of the model to reach 16k). The BRUGEscore shows a harmonic mean between the Rouge2 score and BERT score, giving an equilibrium between sentence semantics and exact 2-gram matching. Lead-1000 is a strong benchmark in this task. These results indicate the superiority of transformer-based summarisation over unsupervised methods and reinforcement learning.

We should mention that Embedding-based metrics compare the semantics between the system summary and the reference summary, whereas rouge metrics will compare n-gram variants. It means Rouge could potentially penalise some summaries (e.g. rouge1), especially when we have several reference summaries. It is difficult to match all the standard gold summaries with comparable scores. In addition, Rouge can not detect synonyms, which means it will penalise two sentences having the same meaning expressed with different words. Moreover, we can see that Bleurt is not a suitable metric to evaluate a system's performance since it generates negative results. However, it could be used to rank different systems only[13].

Furthermore, we can see the impact of generated tokens by the model on the accuracy of matching a good summary candidate. Generating more tokens does not mean to match exactly a good system summary. Therefore, finding the optimal number would maximise the evaluation score. We can also see the impact of dataset preprocessing. Making an objective training process by creating different training pairs for every input annual report (if the report has five gold standards, we would create five training pairs) does not explicitly increase the result and may increase the computational need exponentially. Finally, we can easily distinguish that larger models and models that can accept a higher maximum input token numbers would outrank other models.

6.7.3 Visualisation and Exploration of Results

In this section, we analyse the results by plotting different graphs (*boxplots*, *DistPlots* and *CorrPlots*) in order to provide more detail on our results.

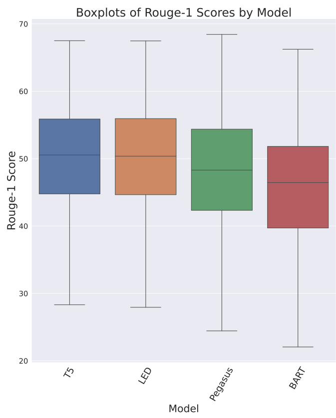
Box Plots: Figure 6.4 shows the boxplots of different evaluation metric scores on the FNS Test Set. Boxplots show the minimum score, first quartile, median, third quartile, and maximum score. The plotted metrics are Rouge1, Rouge2, Rouge3, RougeL, RougeSU4, RougeS1, BERTscore1 (roberta-large-mnli), BERTscore2 (bart-large-mnli) and Bleurt Score. For each metric, we plot the distribution of the individual scores for four transformers: T5 (blue), LED (Orange), Pegasus (Green) and BART (Rouge). On the one hand, We can see that all the variants of ROUGE present comparable boxplots where most scores are condensed around the median.

[13] `https://github.com/google-research/bleurt/issues/1#issuecomment-627615821`

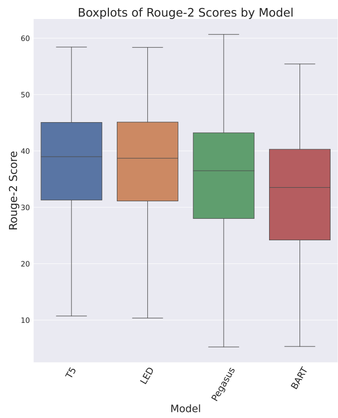
On the other hand, we can see that BERTscores are not normally distributed. They have a left-tail skewed distribution towards the 100 scores (sometimes, the second quartile is the same as the maximum score). Also, the distance between the first and second quartile is higher for bertscores than for rouge scores. The less distributed our evaluation metric is over the median, the more our models behave in the same manner for several annual reports.

Correlation Plots: Figure 6.5 shows the correlation matrices (Pearson R) of different evaluation metrics' scores using summaries produced by different pre-trained transformer models (T5: figure 6.5a, LED: figure 6.5b, Pegasus: figure 6.5c, BART: figure 6.5d) on the FNS test dataset. The correlation plots show that the different variants of the ROUGE metric are highly correlated, which motivates the use of only one ROUGE variant in the evaluation process. In addition, plots show that BERTScore1 and BERTScore2 are highly correlated. There is also a low correlation between Bleurt and BERTscores, although a low correlation in itself is not necessarily problematic. Furthermore, embedding-based metrics are not correlated with the different variants of ROUGE, which motivates using both categories of metrics to have a more robust evaluation.

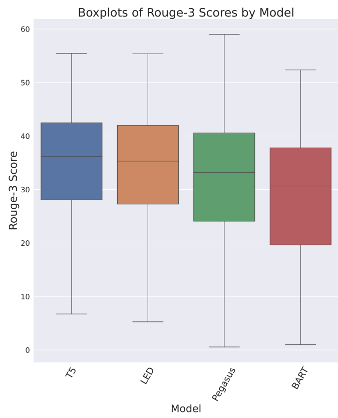
Distribution plots: Figure 6.6 shows the distribution plots of different evaluation metric scores using summaries produced by different pre-trained transformer models (T5, LED, Pegasus, BART) on the FNS test dataset. The Y-axis shows the Share of documents, while the X-axis shows the different evaluation metrics (the sum of the Y-axis value should sum up to 1). We use different scales for every subplot. We present the mean, median, upper and lower quartiles represented using different vertical lines and colours. These plots show that all the evaluation metrics do not show a Gaussian distribution (these findings will be useful in the next section). However, BERTscores show a more left-skewed distribution. Metrics behave similarly for the four reported language models.



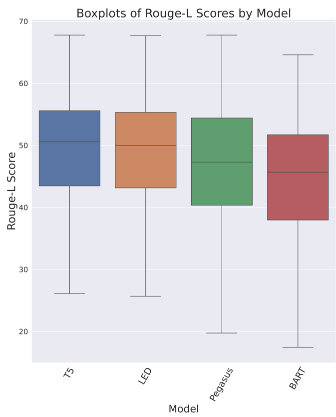
(a) Rouge1



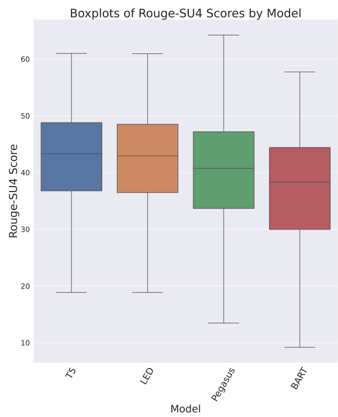
(b) Rouge2



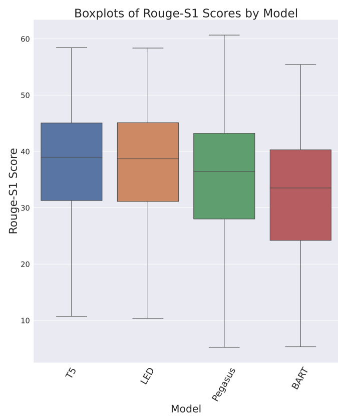
(c) Rouge3



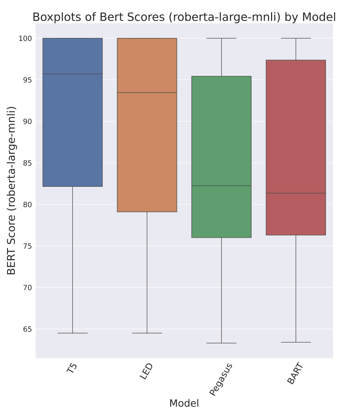
(d) RougeL



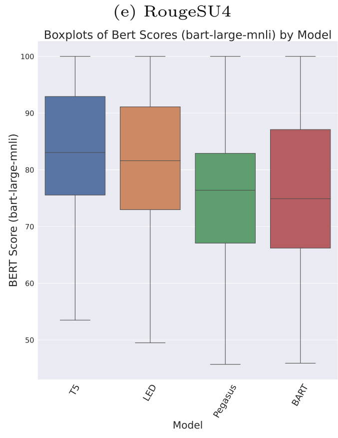
(e) RougeSU4



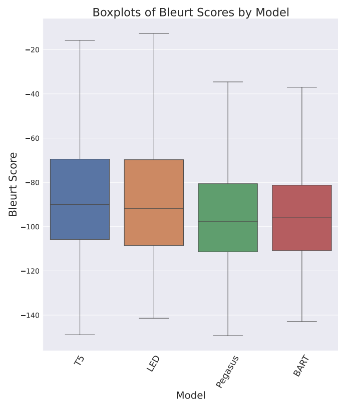
(f) RougeS1



(g) BERTscore1 (roberta-large-mnli)

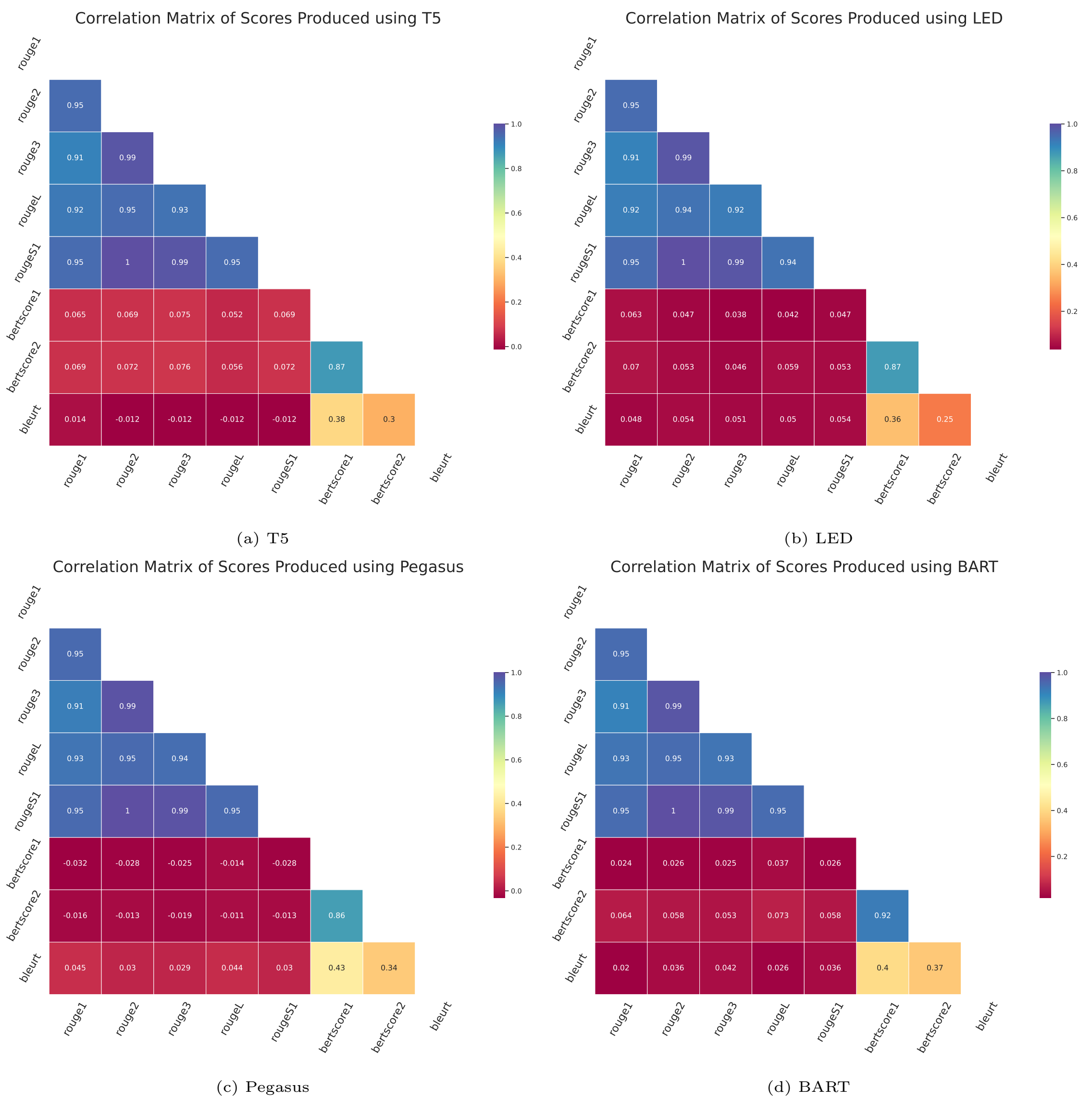


(h) BERTscore2 (bart-large-mnli)



(i) Bleurt Score

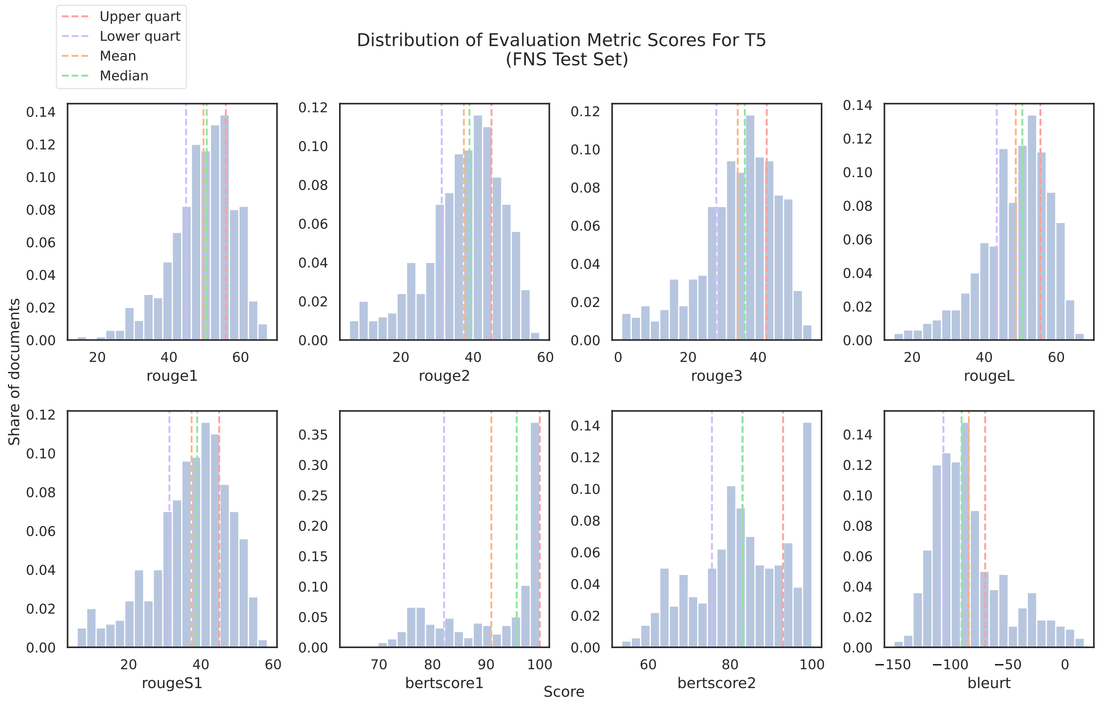
Figure 6.4: Boxplots of Different Evaluation Metrics scores using summaries produced by different pretrained transformer models (T5, LED, Pegasus, BART) on the FNS test dataset



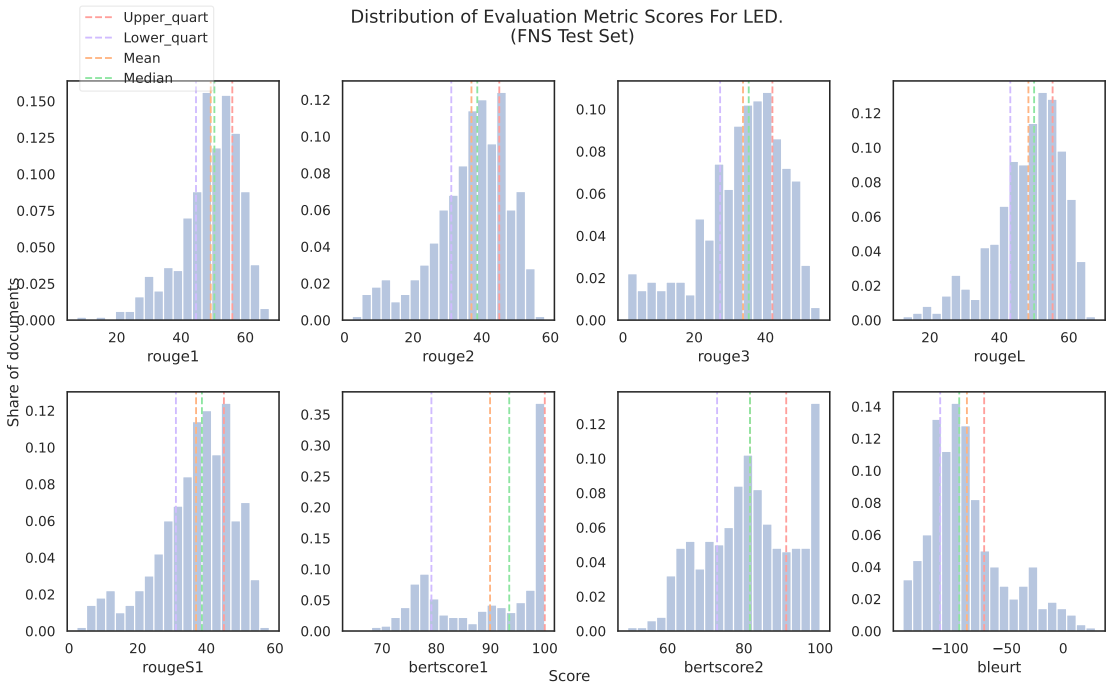
(a) T5 (b) LED

(c) Pegasus (d) BART

Figure 6.5: Correlation matrix (Pearson r) of different evaluation metrics scores using summaries produced by different pretrained transformer models (T5, LED, Pegasus, BART) on the FNS test dataset

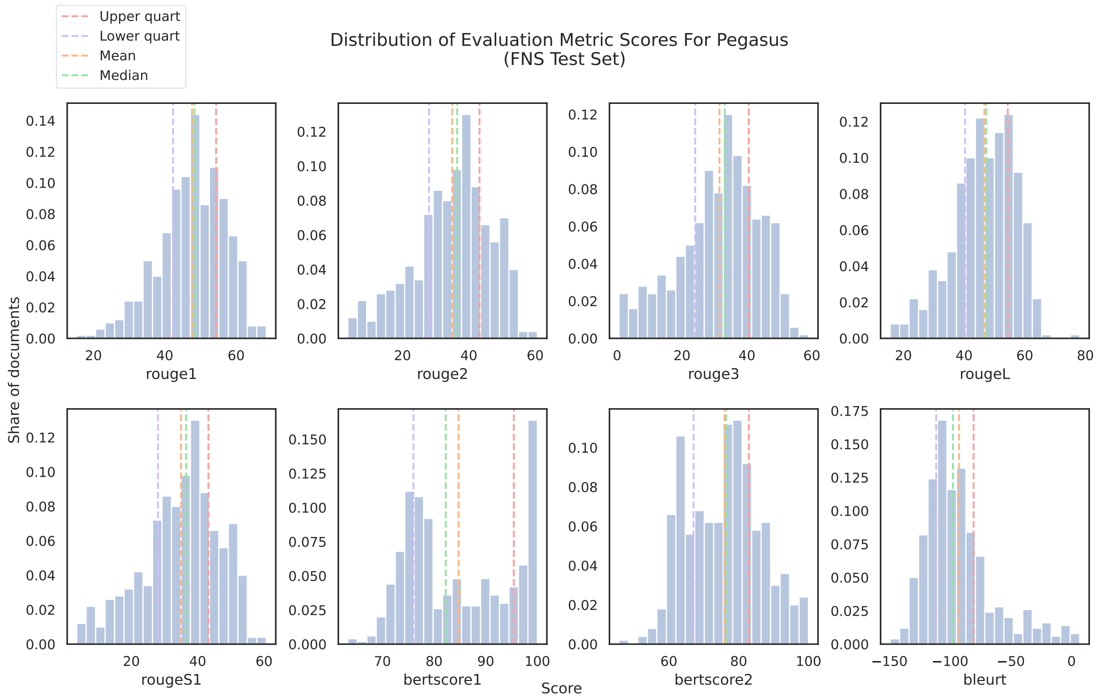


(a) T5

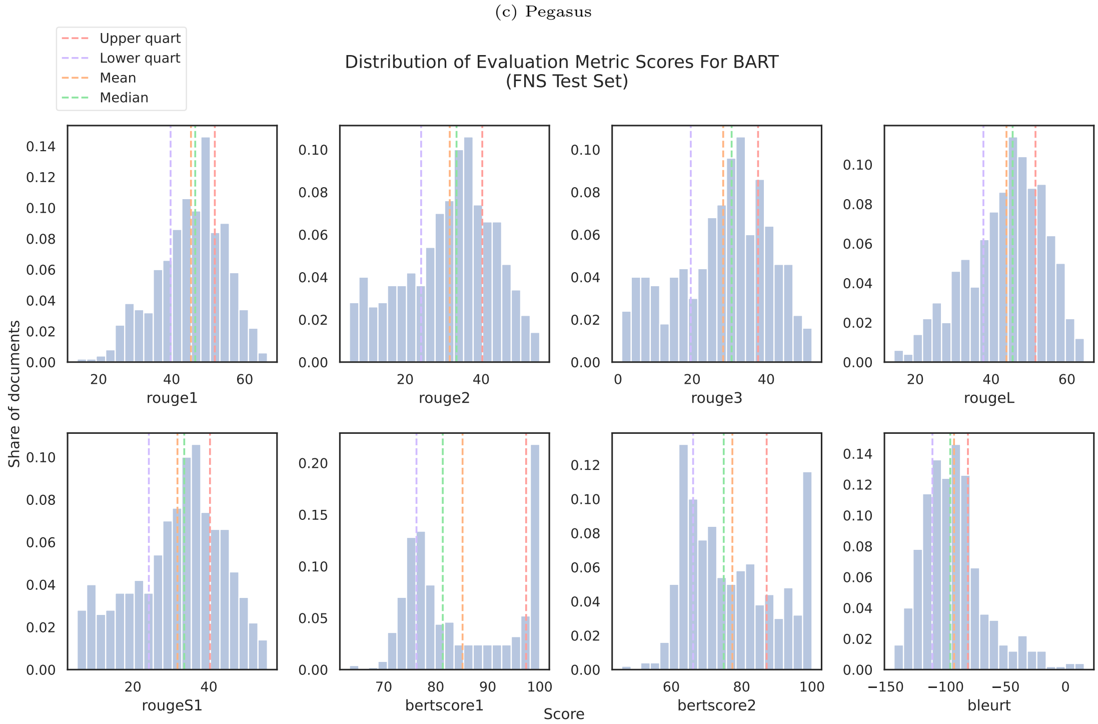


(b) LED

Figure 6.6: Distribution Plots of Different Evaluation Metrics scores using summaries produced by different pretrained transformer models (T5,LED,Pegasus,BART) on the FNS test dataset. The mean, median, upper and lower quartiles are annotated onto the figures.



(c) Pegasus



(d) BART

Figure 6.6: Distribution Plots of Different Evaluation Metrics scores using summaries produced by different pretrained transformer models (T5,LED,Pegasus,BART) on the FNS test dataset. The mean, median, upper and lower quartiles are annotated onto the figures.

6.8 Statistical Significance Testing for NLP

6.8.1 Hypothetical Testing

To compare the performance of two algorithms or models or systems, we need to prove that the evaluation metric, denoted by 'e', is greater for one system than the other. If system A produces an output o_A and system B produces an output o_B, we should prove that e(o_B) > e(o_A) if we would like to prove that algorithm B is better than algorithm A .However, this is not sufficient as we also need to check the statistical significance of the difference in performance between the two models/systems. In this case, the common practice in NLP is to claim superiority of one model over another only if the difference in results is statistically significant. To do that, we had to choose and use a significance level and calculate p-values to determine whether the test results are statistically significant, to avoid false discoveries from evaluating an NLP models and make sure that the probability of making a false claim is minimal. In other words, we form a statistical test to evaluate whether an NLP model can generate "true" out of sample predictions. This is motivated by the fact that the difference between the performance of two algorithms or models could be coincidental. We want to show that this is not the case. To do so, we follow (Dror et al., 2018) and we model our problem as a "no difference" (null hypothesis H_0) or "difference" (alternative hypothesis H_1) and we can choose a parametric or non parametric statistical test depending on the distribution of the test sample. Following (Dror et al., 2018) the problem could be formulated as comparing the performance of two models, A and B, applied on a dataset X, using an evaluation measure M (Eg. Rouge-2 or Rouge-L) (see equation 6.3).

$$\delta(x) = M(R(x)) - M(O(x)) \tag{6.3}$$

Figure 6.7 from (Dror et al., 2018) gives a clear process on how to choose the statistical test to use in NLP applications.

6.8.2 Testing Process

In the first step, we tested the normality of the data using Shapiro-Wilk test[14] (following figure 6.7). The test showed that our data is not normally distributed, and all the summarisation metrics (ROUGE-2, ROUGE-L, ROUGE-S1, Bertscore('Roberta-large-mnli') and Bleurt score) generate non-normal data results. This was aligned with the assumptions presented by Dror et al. (2018), where the authors state that the distribution of F1 scores of Rouge results cannot always be assumed to be gaussian[15]. Therefore, we can not use the classical powerful parametric tests (student's t-test) (Kalpić et al., 2011; Kim, 2015; Mishra et al., 2019). So, we

[14] https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test

[15] The authors of the paper (Dror et al., 2018) presented an annexe document guide that helps to choose the statistical test for different NLP tasks https://arxiv.org/pdf/1809.01448.pdf

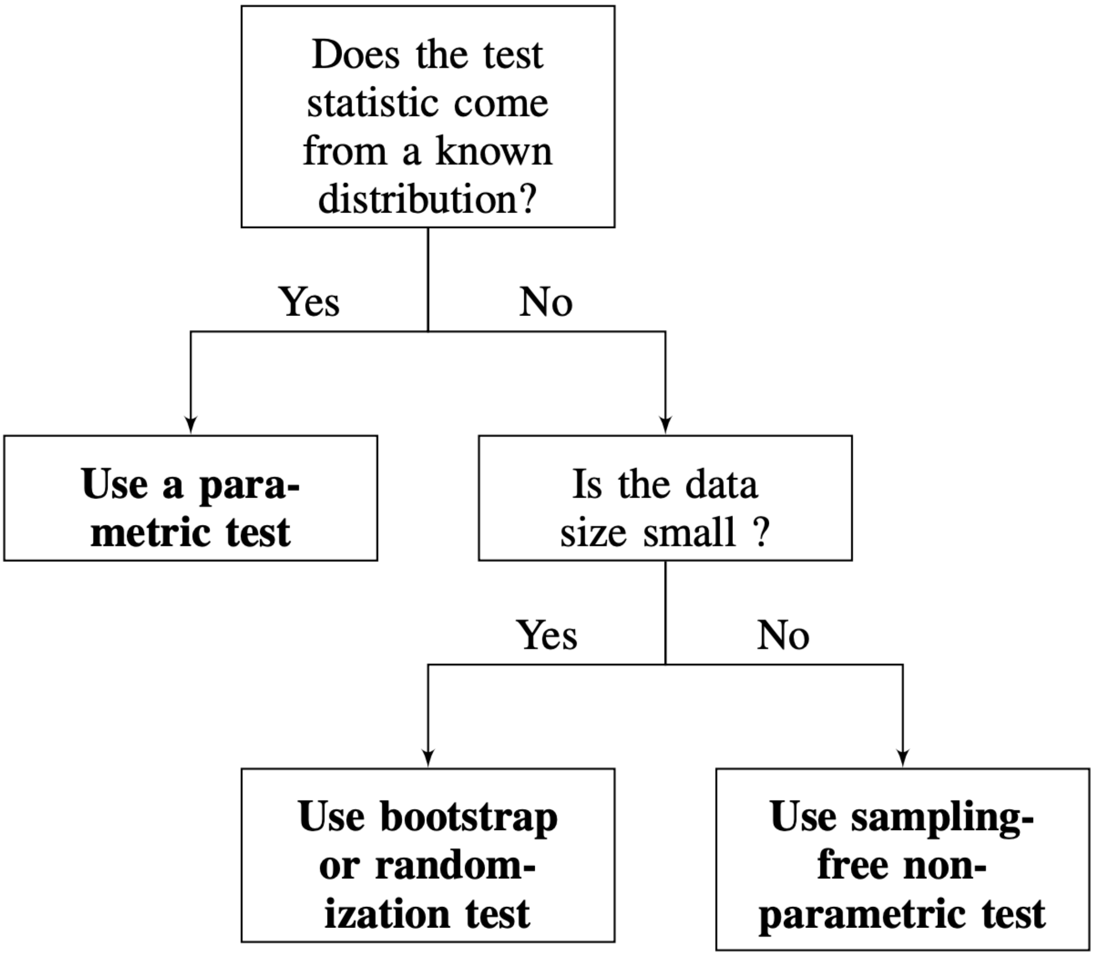


Figure 6.7: Decision tree for statistical significance test selection. Source (Dror et al., 2018)

should opt for a Non-Parametric test, which does not assume anything about the test statistic distribution. Hence, we choose the non-parametric paired bootstrap test (Efron et al., 1993) (see algorithm 4 that demonstrates how bootstrap approximates the p-values) to verify the significance of our results as recommended by the guide (Dror et al., 2018). Paired Bootstrap test was previously used with a variety of NLP tasks such as text summarisation and machine translation (Koehn, 2004; Ouchi et al., 2017; Wu et al., 2017). We apply our test to the difference between the series of results generated by each system. We model our problem as "no difference" (null hypothesis H0) or "difference" (H1). This is a "two-tailed" statistical test.
H_0: There is no significant difference between the two systems' scores on a given population.
H_1: The alternative hypothesis is that there is a significant difference (abnormal difference) between the two systems' scores on a given population.

Or we can formulate the hypothesis this way: $H_0 : \mu_1 - \mu_2 = 0$ vs $H_1 : \mu_1 - \mu_2 \neq 0$ (two-tail), where μ_1 represents the scores of system 1 and μ_2 represents the scores of system 2

Technically, we want to reject the null hypothesis H_0 and accept the alternative hypothesis H_1. To do so, we had to prove that equation 6.4 or equation 6.5 is true for some given α. In other terms, we should avoid the two tails of the distribution and be in the significance region. Following (Søgaard et al., 2014), we can assume that our test sample of 500 elements (size of the test split) is enough to ensure the type 1 errors[16] are minimised.

The output of our analysis is the p-value of the statistical test and a statement on the significance of the difference between the two models. To interpret the results, we compare the p-value to the selected significance level (alpha=0.1). If the p-value is less than the significance level, we know that the test statistic fell into the critical region.

$$p(\delta(X) \leq 0) \leq \frac{\alpha}{2} \tag{6.4}$$

$$p(\delta(X) \geq 0) \leq \frac{\alpha}{2} \tag{6.5}$$

[16]Type 1 error refers to the case where we reject a true null hypothesis. The probability of a type 1 error is called the level of significance of the test or α. Type 1 error: is rejecting H_0 when H_0 is true
Type 2 error refers to the case where we fail to reject a false null hypothesis. The probability of a type 2 error is denoted β. type 2 error: not rejecting H_0 with H_0 is false

ALGORITHM 4: Bootstrap Algorithm

Input: Data sample A and B, n : sample size, R: number of bootstrap repetitions
Output: P-value of the Bootstrap test

$\delta_{orig} \leftarrow \frac{\sum_{i=1}^{n}(A_i - B_i)}{n}$;
$r \leftarrow 0$;
for *x in range(*0, *R)* **do**
    $temp_A \leftarrow []$; ; // empty list
    $temp_B \leftarrow []$; ; // empty list
    samples $\leftarrow$ generate random integers with replacement from 0 to n; (which samples to add to the subsample with repetitions).
    for *sample in samples* **do**
        $temp_A$.append(A_{sample});
        $temp_B$.append(B_{sample});
        $\delta \leftarrow \frac{\sum_{i=1}^{n}(temp_A[i] - temp_B[i])}{n}$;
        if $(\delta > 2\delta_{orig})$ **then**
            $r \leftarrow r + 1$;
        end
    end
end
$pval \leftarrow \frac{r}{R}$;
return *pval*;

6.8.3 Results (p-values) and Analysis

This section is dedicated to presenting the results of the statistical testing that we performed using the bootstrap testing method (Efron et al., 1993). Tables 6.9, 6.10, 6.11, 6.12, 6.13 report respectively the p-values performed on the ROUGE-2, ROUGE-L, ROUGE-S1, Bertscore('Roberta-large-mnli') and Bleurt score. Technically, we apply the bootstrap test to the difference between the series of results generated by the two systems that we would like to compare.

The results reported in the above-mentioned tables report p-values. These values, compared to the significant level α (0.1), determine whether the difference in performance of the two systems is significant or not. If it is not significant, the cell is coloured in red (p-value greater than the significance level). If the cell is not coloured (p-value smaller than the significance level), we can claim with 90 % confidence that there is a significant difference in performance between system A and system B using the chosen metric. The tables should be read vertically where the first cell in the first column shows the p-value of the statistical test performed between system 1 and system 2 (systems are ordered following the used metric in each test). Globally, our results show that most of the differences between two different system summarisers are statistically significant except for a few cases. We note that the non-significant scores happen in the case of two consecutive system summaries, which is a completely logical result.

T5-Small-96																				
LED-BASE-128	0.1243																			
LED-BASE-256	0.0777	0.1649																		
LED-BASE-1000	0.0453	0.0874	0.2068																	
PEGASUS	0.0000	0.0000	0.0000	0.0000																
T5-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0003															
T5-Small-256	0.0000	0.0000	0.0000	0.0000	0.0001	0.2906														
PEGASUS-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.1807	0.3979													
T5-Small-512	0.0000	0.0000	0.0000	0.0000	0.0000	0.0524	0.2062	0.2638												
BART	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0023	0.0089	0.0187											
BART-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0021	0.2190										
LED-LARGE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0526	0.1431									
LEAD-1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0273	0.0724	0.4301								
RNN-LSTM-RL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000							
MUSE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000						
LSA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000					
SBERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3925				
BERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0070	0.0002			
LEXRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
POLY	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
TEXTRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 6.9: pvalues on the Rouge-2 score results using the Bootstrap test. Every column i will include the pvalues of system i and the remaining n-i systems

T5-Small-96																				
LED-BASE-256	0.1075																			
LED-BASE-128	0.1330	0.7523																		
LED-BASE-1000	0.0520	0.1245	0.0740																	
PEGASUS	0.0000	0.0000	0.0000	0.0000																
T5-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000															
PEGASUS-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.2039														
T5-Small-256	0.0000	0.0000	0.0000	0.0000	0.0000	0.2694	0.5332													
T5-Small-512	0.0000	0.0000	0.0000	0.0000	0.0000	0.1049	0.3750	0.3654												
BART	0.0000	0.0000	0.0000	0.0000	0.0000	0.0010	0.0138	0.0086	0.0159											
BART-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.0015	0.2342										
LEAD-1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0220	0.0375									
LED-LARGE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0483	0.1052	0.5720								
RNN-LSTM-RL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0154	0.0451	0.4762	0.4233							
MUSE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000						
LSA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000					
SBERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000				
BERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000			
LEXRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0031		
TEXTRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
POLY	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 6.10: pvalues on the Rouge-L score results using the Bootstrap test. Every column i will include the pvalues of system i and the remaining n-i systems

T5-Small-96																				
LED-BASE-128	0.1203																			
LED-BASE-256	0.0839	0.1708																		
LED-BASE-1000	0.0445	0.0844	0.2003																	
PEGASUS	0.0000	0.0000	0.0000	0.0000																
T5-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0003															
T5-Small-256	0.0000	0.0000	0.0000	0.0000	0.0003	0.2951														
PEGASUS-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.1822	0.4041													
T5-Small-512	0.0000	0.0000	0.0000	0.0000	0.0000	0.0537	0.1971	0.2650												
BART	0.0000	0.0000	0.0000	0.0000	0.0000	0.0009	0.0020	0.0101	0.0215											
BART-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0026	0.2218										
LED-LARGE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0559	0.1457									
LEAD-1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0292	0.0727	0.4374								
RNN-LSTM-RL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000							
MUSE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000						
LSA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000					
SBERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3790				
BERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0076	0.0000			
LEXRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
POLY	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
TEXTRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 6.11: pvalues on the Rouge-S1 score results using the Bootstrap test. Every column i will include the pvalues of system i and the remaining n-i systems

T5-Small-96																				
LED-BASE-128	0.0448																			
LED-BASE-256	0.0161	0.3352																		
LED-BASE-1000	0.0042	0.1930	0.3210																	
BART	0.0000	0.0000	0.0000	0.0000																
PEGASUS	0.0000	0.0000	0.0000	0.0000	0.2440															
T5-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.2373	0.4825														
T5-Small-256	0.0000	0.0000	0.0000	0.0000	0.0106	0.0378	0.0351													
T5-Small-512	0.0000	0.0000	0.0000	0.0000	0.0015	0.0112	0.0059	0.2900												
PEGASUS-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001											
BART-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0077										
LSA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0550	0.0000									
SBERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0300	0.0000	0.2948								
LEAD-1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0093	0.0172							
LED-LARGE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0068	0.0132	0.4912						
BERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1234	0.1335					
RNN-LSTM-RL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001	0.0000				
MUSE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0329			
LEXRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
TEXTRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0423	
POLY	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.1001

Table 6.12: pvalues on the BERT score results using the Bootstrap test. Every column i will include the pvalues of system i and the remaining n-i systems

T5-Small-96																				
LED-BASE-128	0.2622																			
LED-BASE-256	0.1766	0.3579																		
LED-BASE-1000	0.0774	0.2204	0.3232																	
PEGASUS	0.0000	0.0001	0.0004	0.0015																
BART	0.0000	0.0000	0.0003	0.0011	0.4284															
LSA	0.0000	0.0000	0.0000	0.0000	0.1042	0.1160														
SBERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0010	0.0006	0.0134													
T5-Small-256	0.0000	0.0000	0.0000	0.0000	0.0020	0.0013	0.0142	0.4225												
T5-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0014	0.0008	0.0113	0.3947	0.4955											
PEGASUS-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0008	0.0009	0.0015	0.3410	0.4104	0.4304										
BERT-EXTRACTIVE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1452	0.2396	0.2400	0.3016									
T5-Small-512	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0233	0.0598	0.0539	0.0906	0.1589								
RNN-LSTM-RL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0004	0.0002	0.0005	0.0008	0.0177							
LEAD-1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0005	0.1594						
LED-LARGE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	0.1439	0.4598					
BART-MULTI-REFERENCES	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0006	0.1427	0.4502	0.4757				
MUSE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0586	0.3071	0.3319	0.3543			
LEXRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0164	0.1609	0.1706	0.1829	0.2828		
POLY	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0015	0.0284	0.0433	0.0390	0.0723	0.1871	
TEXTRANK	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0014	0.0004	0.0018	0.0026	0.0171	0.0946

Table 6.13: pvalues on the Bleurt score results using the Bootstrap test. Every column i will include the pvalues of system i and the remaining n-i systems

6.9 Adversarial Analysis

To assess the robustness of the metrics, we also performed an adversarial analysis on the predicted summaries. Adversarial attacks on an NLP model are text perturbations designed to test the effectiveness of the metrics. Our experiments involved corrupting a set of system summaries generated by the T5 small model, which was the best-performing model on the test dataset. The aim is to evaluate how well the metrics reflect the difference between the corrupted and uncorrupted text. We tested the ability of the metrics to resist different sources of noise using BERT mask-filling, word-dropping, and word permutation methods. Theoretically, corrupting a text will degrade its quality; the original text should get a higher score than the corrupted one. Therefore, we can assess the robustness of our metrics and their ability to resist different sources of noise.

Data: we decided to perform our adversarial analysis only on the T5 small model (the best-performing model on the test dataset). This is motivated by the hypothesis that the higher the quality of the summary, the higher the probability that noise will make it worse.

Corruption Methods: our chosen methods of corruption are detailed as follows:

- **Word-dropping** corrupts the summary by deleting tokens. This will simulate some of the common issues that may happen with extractive summarisation. Technically, we will mask random words in every chunk and then drop these masked words. (see algorithm 5)

- **BERT mask-filling** is a denoising encoding task. Some of the input tokens are masked, and a pre-trained BERT model, such as Roberta will predict and infill these tokens. This introduces new words in the sentence but keeps the general sense of the sentence. BERT mask-filling is very challenging for BERT score because it supposes that the predicted word by a BERT model is better in this context rather than the original word in the system summary. (see algorithm 6)

- **Word Permutation** tests the metrics' sensitivities to correct syntax by swapping the ordering of two adjacent words/tokens in the summary. This metric will penalise the n-gram-based metrics except rouge-1. However, it will favour model-based metrics (E.g. BERT score). Technically we will mask a random token in a every chunk and then we swap the following or the previous two tokens. (see algorithm 7)

ALGORITHM 5: Performing random word dropping. This drops a token from each chunk

Data: chunk of length n tokens
Result: Corrupted chunk using the word dropping technique

```
begin
    1- preprocessing step (Mask random word in a chunk):
    for chunk in chunks do
        mask_index = randint(0, len(chunk)); chunk[mask_index] = MASK;
    end
    return Masked chunks
    2- Corrupt: Drop masked words:
    corrupted_chunks = [s.replace(self.MASK, ”) for s in self.inputs]
    return Corrupted chunks
    3- Postprocess: Joins sequences of chunks so it can be written to a file.
end
```

ALGORITHM 6: Performing BERT Mask-filling: This masks random tokens in the summary and uses a pre-trained BERT to in-fill these.

Data: chunk of length n tokens
Result: Corrupted chunk using the Bert Mask Filling

```
begin
    1- preprocessing step (Mask random word in a chunk):
    for chunk in chunks do
        mask_index = randint(0, len(chunk)); chunk[mask_index] = MASK;
    end
    return Masked chunks
    2- Corrupt: Bert Mask Filling: Feed through pipline to predict masked words ”’
    3- Postprocess: Joins sequences of chunks so it can be written to a file.
end
```

ALGORITHM 7: Performing random word permutation: switching the ordering of two adjacent tokens for each chunk

Data: chunks
Result: Corrupted chunk using the word permutation technique

```
begin
    1- preprocessing step (Insert mask token before tokens to be permuted):
    for chunk in chunks do
        if len(chunk) ≥ 2 then
            mask_index = torch.randint(len(chunk)-1, (1,)).item();
            chunk.insert(mask_index, self.MASK);
        end
    end
    return chunks;
    2- Corrupt: Permute two adjacent tokens after the ⟨MASK⟩:
    3- Postprocess: Joins sequences of chunks so it can be written to a file.
end
```

In order to avoid the creation of a bias in the distribution of corrupted tokens, we opted for a uniform distribution of noise which means that we will divide the text into chunks, and then we will apply one corruption every chunk. We choose four values of chunks: 4, 6, 8 and 10. The higher the chunk length is, the fewer corrupted tokens per text file we will have. By uniformly distributing the corruption across the text, we can evaluate how well the metrics reflect the difference between the corrupted and uncorrupted text. The listings 6.2 and 6.3 show two examples of corrupted text.

```
{
"sentence": "We work closely with our OEM partners, often for several
    years before a new model is launched, providing support throughout
    the planning process with a precise view of  market pricing, local
    fit requirements and sales volumes.",
"Word permutation": "We work closely our with OEM partners, often
    several for years a before new model is providing launched, support
     throughout the planning process a with view precise of
market pricing, fit local requirements and volumes sales.",
"Word dropping": "We closely with our OEM partners, often for years
    before a model is  providing support
throughout the planning process with  precise of market pricing, local
    requirements and sales",
"Bert Mask Filling": "We work closely with our OEM Very often for
    several years before a new model is for providing support
    throughout the planning process as a precise view    market pricing
    , local fit requirements. Volume volumes."
}
```

Listing 6.2: Example of three Different Corruption Techniques (Example randomly selected)

```
{
"sentence": "During 2009, we will reinforce our leadership in hedge
    fund investing with the launch of an integrated hedge fund
    management business to improve performance, adapt to changing
    investor demands and leverage our global scale. ",
"Word permutation": "2009, During we will reinforce our leadership in
    fund hedge investing with the launch an of integrated hedge fund
    management to business improve adapt performance, to changing
    investor demands and leverage global our scale.",
"Word dropping": "During 2009, will reinforce our  in hedge fund
    investing
with launch of an integrated hedge fund management to improve
    performance, to changing investor demands and our global scale.",
"Bert Mask Filling": "In 2009, we will reinforce our Invest in hedge
    fund investing announces the launch of an integrated hedge fund
    management business to improve performance, adapt to changing
    market demands and expanding our global scale."
}
```

Listing 6.3: Example of three Different Corruption Techniques (Example randomly selected)

Results and Analysis of Adversarial Attack:

To compare the original and corrupted summaries, we use a strict comparison where the original summary must be strictly better than the corrupted one. Tables 6.14, 6.15, 6.16 and 6.17 shows the results for the three adversarial attacks (mean accuracy per metric) with a chunk length of 4, 6, 8 and 10 respectively (All standard deviations were small (less than 0.2%)). The accuracy value calculates the percentage of non-corrupted summaries that received better scores than their corrupted counterparts. Each metric is used to score the original and the corrupted versions of the summaries. This task should give the uncorrupted version a higher score to make sure that the metric is sensitive to corrupted summaries. An accuracy value of 0.00 indicates that the corrupted and non-corrupted summaries received the same scores, as with ROUGE-1 during the word permutation corruption test. Indeed, ROUGE-1 cannot distinguish at all between the corrupted and uncorrupted summaries as it is syntax-insensitive.

BERTScores achieved an accuracy score of around 60% across the three different tasks while different variations of Rouge score achieved high accuracy scores (around 90%). The results also show that ROUGE-2 performed best on the word permutation and BERT mask-filling tasks, while ROUGE-3 performed best on the word-dropping task. In fact, when the corruption is applied to a single token in a sentence, it disrupts the n-gram sequence, which impacts ROUGE-n when n is greater than 1. Bleurt showed poor results, confirming that it is more suitable for comparing different models than evaluating a single model. In addition this proves that language model-based metrics resists less to noise than the n-gram based ones. These results suggest that **ROUGE** is better suited for extractive summarisation while model-based metrics are more suitable for abstractive summarisation. **Rouge** assesses whether the summary is good or not on a word-by-word basis, whereas model-based metrics consider the context as a whole.

Finally, we include tables B.2, B.3, B.4 in appendix B.3 that show the ranking of the original summary against the different corrupted summaries. This confirms that the higher the frequency of injecting corruptions, the lower the evaluation score is. In addition, we see that the word dropping impacts the least the score of a summary while the Bert mask filling impacts the most (negatively) the score of the corrupted summary.

Metric	Word dropping_4 (%)	Word Permutation_4 (%)	Bert Mask filling_4(%)
ROUGE-1	0.878	0.00	0.974
ROUGE-2	0.98	0.998	0.978
ROUGE-3	0.99	0.998	0.978
ROUGE-S1	0.98	0.998	0.978
ROUGE-S2	0.966	0.998	0.978
ROUGE-L	0.954	0.992	0.978
ROUGE-SU4	0.928	1	0.978
BERTScore1	0.622	0.67	0.684
BERTScore2	0.646	0.664	0.768
BLEURT	0.624	0.67	0.644

Table 6.14: Mean accuracy by metric on the three corruption tasks. We apply three types of corruption to the system-generated summaries. We create a corruption every 4 chunks.

Metric	Word dropping_6 (%)	Word Permutation_6 (%)	Bert Mask filling_6(%)
ROUGE-1	0.842	0.00	0.98
ROUGE-2	0.962	1	0.984
ROUGE-3	0.976	0.996	0.982
ROUGE-S1	0.962	1	0.984
ROUGE-S2	0.954	0.998	0.984
ROUGE-L	0.948	0.984	0.982
ROUGE-SU4	0.9	0.994	0.984
BERTScore1	0.63	0.666	0.674
BERTScore2	0.628	0.644	0.714
BLEURT	0.616	0.622	0.582

Table 6.15: Mean accuracy by metric on the three corruption tasks. We apply three types of corruption to the system-generated summaries. We create a corruption every 6 chunks.

Metric	Word dropping_8 (%)	Word Permutation_8 (%)	Bert Mask filling_8(%)
ROUGE-1	0.824	0.000	0.982
ROUGE-2	0.962	1	0.984
ROUGE-3	0.972	0.996	0.986
ROUGE-S1	0.962	1	0.984
ROUGE-S2	0.95	0.998	0.986
ROUGE-L	0.944	0.972	0.984
ROUGE-SU4	0.898	0.988	0.982
BERTScore1	0.592	0.646	0.664
BERTScore2	0.6	0.628	0.686
BLEURT	0.586	0.63	0.582

Table 6.16: Mean accuracy by metric on the three corruption tasks. We apply three types of corruption to the system-generated summaries. We create a corruption every 8 chunks.

Metric	Word dropping_10 (%)	Word Permutation_10 (%)	Bert Mask filling_10(%)
ROUGE-1	0.826	0.000	0.982
ROUGE-2	0.958	1	0.99
ROUGE-3	0.968	0.998	0.992
ROUGE-S1	0.958	1	0.99
ROUGE-S2	0.946	0.996	0.992
ROUGE-L	0.922	0.978	0.99
ROUGE-SU4	0.88	0.994	0.992
BERTScore1	0.608	0.63	0.668
BERTScore2	0.636	0.6	0.656
BLEURT	0.556	0.632	0.574

Table 6.17: Mean accuracy by metric on the three corruption tasks. We apply three types of corruption to the system-generated summaries. We create a corruption every 10 chunks.

6.10 Limitations

To perform the training of a summarisation model, we need a set of gold standard summaries for every input text that we want to summarise. Ideally, a gold standard summary is a summary of the input text which is extracted from the original text or generated by human experts. We believe that the biggest technical challenge that is facing the financial text summarisation research community is the lack of gold standards, specifically human-generated summaries by domain experts (Kennedy et al., 2010). At the moment, we are only using extracted sections from annual reports as gold summaries. Furthermore, although there is a lot of work done for English corpora, other languages still need to be better resourced. Hence, we need to mention explicitly that these results are limited to this English dataset. We do not guarantee the same performance of the evaluation metrics in other languages, especially the model-based evaluation metrics, because they are pretrained in English. In addition, financial data sets are very large and scalable, which implies the necessity of significant computational capacities (e.g. GPUs, TPUs). Another challenge is that the jargon used in financial disclosures is different from the 'general' language. Loughran et al. (2011) showed that the meaning of words can change substantially in a financial context. Therefore there is an urgent need to pre-train financial-specific language models to be used in such studies.

6.11 Ethical considerations

Environmental footprint: Strubell et al. (2019) discussed the impact of progress in hardware and methodology for training neural networks has caused a substantial increase in energy consumption (electricity). In addition, training NLP models has a huge environmental impact due to the carbon footprint required to fuel modern tensor processing hardware. The leading models in text summarisation and NLP have become larger and increasingly data-hungry since the development of the Transformer. The NLP community is becoming more cognisant of the environmental impacts of training these models, which frequently require days of energy-intensive GPUs for a single training cycle (Strubell et al., 2019). In addition, the high energy requirements of these large models is a source of real concern since energy is not yet generated from carbon-neutral sources. For this reason, we started using the Weights Biases compute tracker[17] to measure the GPU compute. In addition, our High-End Computing (HEC) infrastructure offers a detailed report about GPU use after finishing the training. Added to that, we can use the MLCO2 machine learning emissions calculator[18] to evaluate our use of GPU in terms of CO2 emissions.

CO2 Emission Related to Experiments: Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 kgCO_2eq/kWh. A total

[17]Wandb compute tracker library: `https://wandb.ai/`

[18]`https://mlco2.github.io/impact/#compute`

of 34.65 hours of computation was performed on the hardware of type Tesla V100-SXM2-32GB (TDP of 300W). Total emissions are estimated to be 4.49 kgCO_2eq of which 0 percents were directly offset. Estimations were conducted using the MachineLearning Impact calculator presented in (Lacoste et al., 2019). 4.49 kg of CO2 is equivalent to 18.1 Km driven by an average car, 2.25 Kgs of coal burned, 0.07 Tree seedlings sequesting carbon for 10 years.

Disclaimer: Currently, the carbon footprint of model training may not be a major issue compared to other economic activities. However, in the near future, it has the potential to become a major problem, especially with the emergence of very large language models such as chatGPT-4. The rising cost of CPU/GPU/TPU hardware is already a challenge, and it is expected to become more problematic in the future with the increasing price of metals and semiconductors. As a response, some machine learning practitioners are encouraging an open-source culture in the field to avoid wasting valuable resources on repetitive training and hyperparameter optimisation. Such practices may drastically reduce the cost and the environmental impact of NLP and ML research.

Legal issues with NLP corpora: As stated by the Corpus Creation guide [19], copyright issues in corpus creation are complex and unavoidable. In fact, financial data raises several copyright issues. Although a lot of financial qualitative data (e.g. financial reports) is publicly available, we are not allowed to redistribute it in its original PDF format. That is why we opt for using data in a converted text format. Moreover data scraping could raise copyright issues with news websites. Other types of financial data may be premium and not accessible to researchers. It is mainly sold by data providers for financial institutions.

6.12 Conclusion of the Chapter

This chapter tackles and describes the complexity of the task of automatic financial extractive summarisation of UK annual reports. We trained a wide variety of transformer models, and we compared them with different unsupervised baselines. We tried a set of model-based evaluation metrics. This set consists of BERTScore, meteor, Bleurt, and a new proposed metric called BRUGEscore. Most of them use transformer language models to compute contextual embeddings for the input sequence. We analysed the results and performed data analysis on our results to plot the distribution of the scores, the correlation between the metrics, and the statistical significance of the results. Additionally, we have performed adversarial analysis on our system-generated summaries to verify the metrics' robustness.

[19] https://www.lancaster.ac.uk/fass/projects/corpus/ZJU/xpapers/Xiao_corpus_creation.pdf

Chapter 7

Abstractive Summarisation: Use Case on Arabic Financial News Summarisation

In recent years, transfer learning and language models have changed the landscape of NLP with the appearance of models like BERT, BART and T5. These models are setting a new state-of-the-art for several NLU and NLG tasks by allowing for the training of models on large amounts of unlabelled text data. While most of these models are pre-trained on English language corpora and are intended for general language use, we present in this chapter FinAraT5, a text-to-text model for financial Arabic text generation designed for financial use cases.

The financial industry generates a significant amount of multilingual data, and there is a pressing need for better multilingual NLP models for tasks such as summarisation, structure detection, and causal detection in the financial domain. However, there are currently no pre-trained finance-specific Arabic language models available. To address this need, we continue the pre-training of AraT5 model to create FinAraT5, a model trained on a large Arabic financial communication corpus consisting of annual and quarterly reports, press releases and financial newswires.

In this chapter, we hypothesise that FinAraT5 would perform better than other multilingual models in the financial domain, being suited for generative tasks. We demonstrate this through research on a generative task from a novel summarisation dataset called FinAraSum that we presented in section 5.3.2. Our results show that monolingual models are highly competitive with state-of-the-art models such as mT5 BERT, and the original AraT5 on news summarisation task.

7.1 Introduction

The Arabic language is a very rich language with fewer explored language models compared to the English language and other Latin languages. The Arabic language is the sixth spoken language in the world, with over 270 million native speakers[1]. Arabic remains understudied in the NLP community despite the huge effort by the SIGARAB ACL Special Interest Group on Arabic Natural Language Processing[2]. In addition, Arabic Natural Language Processing and generation tasks have proven to be very challenging to tackle. Most Arabic language models are mainly encoder only and are not field-specific (Antoun et al., 2020). There is a growing interest in Arabic language processing due to its widespread use. Arabic language could be divided into Classical Arabic, Modern Standard Arabic as well as several dialectical Arabic variations primarily employed by native speakers for online communication, especially on social media platforms, where the language tends to be less formal. Modern Standard Arabic is used not only in news and media but also in various other formal contexts, such as formal speeches, literature, official documents, legal proceedings, and educational materials. It serves as a standardised form of the language that allows for better mutual understanding among speakers from different Arabic-speaking regions. The main features of the Arabic writing system are the absence of capitalisation and the use of diacritics instead of vowels. Diacritics are not used in most of the Arabic text available online, which makes it difficult to disambiguate meanings even for a native speaker (El-Haj et al., 2016). Arabic exhibits distinct sentence structure and grammar when compared to English and other Latin languages, as it follows a Verb-Subject-Object (VSO) order, unlike the Subject-Verb-Object (SVO) order seen in English. Additionally, Arabic features a rich and notable derivational morphology, setting it apart from many other languages. Another distinctive characteristic of Arabic is its right-to-left writing system, unlike left-to-right written languages like English. This chapter focuses on the use of Modern Standard Arabic (MSA) in a professional context, specifically in finance and business news.

7.2 Motivation

The work in this chapter centres on the Middle Eastern stock exchange markets, where Arabic serves as the predominant communication language. These exchanges have been witnessing a notable surge in market capitalisation, driven by various factors, including the presence of prominent oil and gas companies, flourishing real estate enterprises, and particularly thriving investment firms like Kingdom Holding[3]. Therefore, the Middle Eastern markets are gaining in popularity among Western investors, especially with the evolution of jurisdiction in the UAE through the free trade zone and the flexibility of investment in a Gulf-listed company. In addition, Tadawul Saudi Exchange (Tadawul) is the ninth most significant stock market among the 67 members of the World Federation of Exchanges by market capitalisation of listed companies (approximately US$2.6 trillions on 30 June 2021)

[1] https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/
[2] https://www.sigarab.org/
[3] https://kingdom.com.sa/

and is the dominant market in the Gulf Cooperation Council (GCC). Tadawul, ranked as the third largest stock market among its emerging market counterparts, holds affiliations with prestigious organisations such as the International Organisation of Securities Commissions (IOSCO), the World Federation of Exchanges (WFE), and the Arab Federation of Exchanges (AFE) and it is also included in FTSE Russell and S&P Emerging Market indices. Another critical development that signified the growing maturity of the Saudi market was the introduction of derivatives, which is a significant step to further advance the Saudi capital market by providing investors with a complete and diversified range of investment products and services. These credentials indicate a substantial growth in textual financial data in Arabic, necessitating advancements in Arabic NLP capabilities, especially concerning finance and investment-related tasks. To address this, we propose the development of a specialised monolingual Arabic T5 model tailored to handle financial corpora efficiently. In this chapter, we introduce FinAraT5, an extension of AraT5, achieved through further pre-training on a vast collection of monolingual financial Arabic data. Unlike previously released versions of Arabic BERT, FinAraT5 is designed to excel on generative tasks. Subsequently, we assess the performance of this pre-trained model in financial news summarisation using a newly curated Arabic news summarisation dataset called FinAraSum, which we compiled ourselves. Our research chapter makes the following contributions:

- We describe the steps to preprocess and clean a financial narratives corpus covering different Middle Eastern stock exchange markets (Saudi exchange, Dubai exchange, Abu Dhabi exchange ...).
- We present a pre-trained Arabic text-to-text financial language model pre-trained on a corpus of Arabic financial narratives. The model features 220 million parameters and is trained on 25 GB of PDF text file. The training process took 45 days, utilising a Google Cloud TPU V3.8 for computational power.
- We compare FinAraT5 with different versions of multilingual T5 to prove the importance of training monolingual language models. We show the importance of what we call language-adapted pretraining language models compared to multilingualism.

7.3 Background

7.3.1 Abstractive Summarisation

Sequence-to-sequence architecture is widely considered one of the most effective approaches for abstractive models (See et al., 2017). These abstractive models have proven to be highly efficient for tasks such as news summarisation and text paraphrasing (Çelikyılmaz et al., 2018). Their ability to generate concise and coherent summaries or rephrasings makes them valuable tools for various natural language processing applications. Unlike extractive summarisation, abstractive approaches are not restricted to the input words and generates a new text piece (Rush et al., 2015; Chopra et al., 2016; Liu et al., 2015). One issue with the past work targeting Arabic abstractive summarisation is the evaluation of such models on highly extractive datasets. The primary available Arabic extractive datasets are ANT Corpus (Chouigui et al., 2017) and KALIMAT (El-Haj et al., 2013). Therefore, in this study, we prepared our customised highly abstractive financial summarisation dataset to suit the financial model we created. The mathematical formulation of an abstractive summariser is as follows:

$$p(y|x,\theta) = \prod_{i}^{T_y} p(y_i|\{y_0, ..., y_{i-1}\}, x, \theta) \tag{7.1}$$

Where we will have an input sequence of words x = $[x_1, ..., x_n]$ from a fixed length input vocabulary, the aim of abstractive summarisation is to generate a condensed sequence of Ty summary words y = $[y_1, ..., yTy]$.

7.3.2 T5 transformer

T5 (Raffel et al., 2020) (Text-to-Text-Transfer-Transformer) is a sequence-to-sequence language model based on the transformer architecture developed by Vaswani et al. (2017) and pre-trained on a multi-task mixture of unsupervised and supervised tasks for which each task is converted into a text-to-text format. T5 model is highly versatile and adaptable to a wide range of Natural Language Processing (NLP) tasks. It can perform well on various tasks by using a technique called "prepending a different prefix to the input corresponding to each task." This means that before feeding the input text to the T5 model, they add a specific prefix that indicates the nature of the task they want the model to perform. For translation tasks: They would add a prefix like "translate English to German: [input sentence]" before passing the input sentence through the T5 model. For summarisation tasks, They would add a prefix like "summarize: [input text]" before passing the input text through the T5 model. It is configured for 4096 maximum input tokens where the input and output are always strings. However, the model is based on relative position embeddings, which allows it to scale to longer input sequences (Raffel et al., 2020). Because of the complexity $O(n^2)$ of the Transformer's self-attention mechanism, such scaling increases memory consumption exponentially. The idea of a unified Transformer framework for different tasks was introduced by Raffel et al. (2020). In fact, the T5 framework treats all generative and discriminative tasks as a text-to-text problems. This enabled a more efficient transfer learning approach. In addition, Google researchers recently extended the T5 model to multilingualism by releasing mT5 (Xue et al., 2021), a multilingual version of T5. In this work, we will also test the portability of mT5 to the Arabic language by exploring its performance on Arabic financial task. The authors of the article "Exploring the Limitations of Transfer Learning in a Combined

Text-to-Text Transformer" conducted a comprehensive empirical investigation to determine the most effective transfer learning methods and utilised their findings to create a new model, T5 (Text-To-Text Transfer Transformer). They also introduced an open-source training dataset called Colossal Clean Crawled Corpus (C4) which was used to pre-train T5, resulting in it becoming the top performer on various NLP benchmarks while still providing flexibility for fine-tuning for particular NLP tasks (Najafi et al., 2022; Mastropaolo et al., 2021). Unlike BERT autoencoding models, T5 is based on text-to-text format, representing both input and output as text sequences. The encoder and decoder stay separated, with each layer divided into "blocks" which contain a self-attention layer and a feedforward network.

Presented herewith is a comprehensive and systematic exposition of the provided information, delineating the rationales underpinning the decision to employ T5 for the study:

- T5 models have both encoder and decoder stacks and use relative positional encoding.
- Text-to-text models apply a prefix to the input sequence that determines the NLP task.
- One of the advantages of text-to-text models is that they use the same hyperparameters for all NLP tasks.
- T5 contain a feedforward network and implements beam search.
- T5 makes sure not to repeat n-grams equal to a fixed number to avoid repetition in generated texts.
- T5 controls the length of the samples with minimum length and maximum length. it also applies a length penalty.

7.4 FinAraT5: Financial Text-to-text Model

Language Modeling is the method of creating a system that is tailored to a certain set of data, usually related to a specific topic. There are two types of LM: Masked Language Modeling and Casual Language Modeling. FinAraT5 is a financial Arabic language model designed for text generation. It is trained using a text-to-text approach. Our model is based on araT5 (Nagoudi et al., 2022a), a pre-trained Arabic text-to-text model. When deciding which model to train for the financial Arabic context, we decided to go for the T5 architecture because it is the only model useful in generative and discriminative tasks simultaneously and with the same performance. Figure 7.1 shows a schema explaining the tasks that the model could perform. The proposed framework allows using the same model, loss function and hyperparameters for any NLP task, such as text summarisation (in blue) machine translation (in yellow), paraphrasing (in green), sentiment analysis (grey) and text classification (orange).

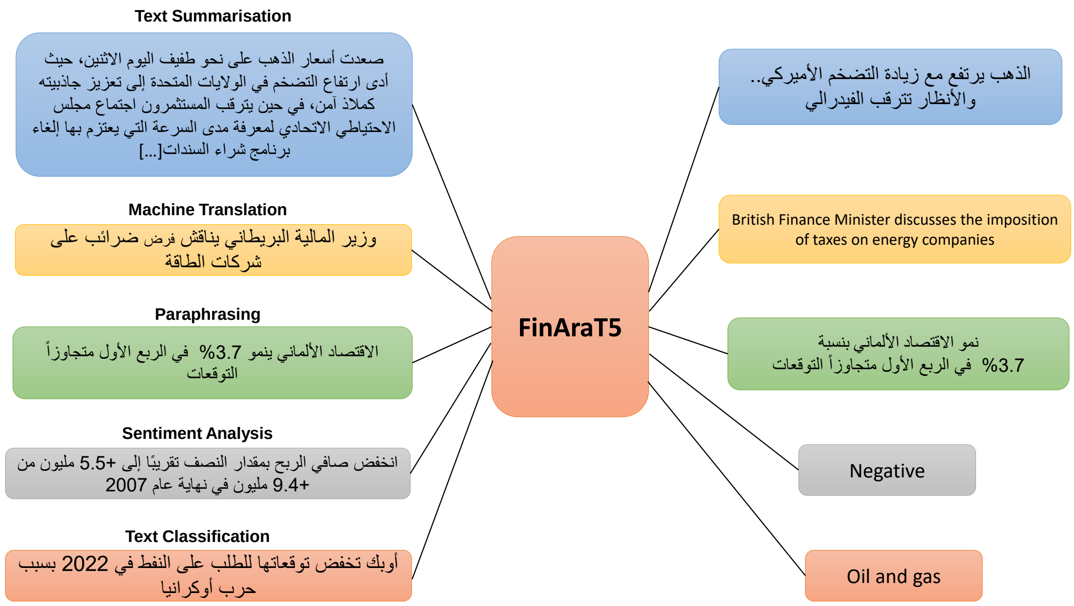


Figure 7.1: FinAraT5: a text-to-text model with some examples of what the mode can do: summarisation, machine translation, paraphrasing, sentiment prediction, text classification

7.4.1 Pretraining a Language Model

Developing a text-to-text model for financial Arabic can be a challenging task and requires several steps. In the following steps, we will describe in general the process of pretraining a language model.

Step 1: Data collection: this is a predominant step in the model training. In fact, without a good dataset, it is impossible to build a good language model. The dataset should be large and diverse enough and should include a variety of different financial documents.

Step 2: Data pre-processing, including text cleaning and normalisation, is a critical step in the pipeline, as datasets may contain valuable information that could significantly impact the performance of models.

Step 3: Loading the dataset and passing it to the model architecture.

Step 4: Training a tokeniser.

Step 5: Initialising a model from scratch.

Step 6: Initialising the trainer.

Step 7: Pre-training the model on the dataset by employing a variety of techniques, such as unsupervised learning, supervised learning, or a combination of both, to establish a strong foundation before fine-tuning for specific tasks.

Step 8: Fine-tune the language model on specific downstream financial NLP tasks, such as text summarisation, sentiment analysis or token classification.

Step 9: Testing the model on a set of unseen financial Arabic textual dataset to evaluate its performance.

7.4.2 Architecture

We use the BASE architecture of T5 encoder-decoder (Raffel et al., 2020), with 12 encoder layers and 12 decoder layers. Both the encoder and decoder have 12 attention heads and 768 hidden units. So, our BASE architecture of FinaraT5 is a text-to-text architecture with both encoder - decoder having 12 layers, 12 attention heads, and 768 hidden units. In total, FinAraT5 Base is an encoder-decoder with 12+12 layers and 220M parameters.[4]

7.4.3 Vocabulary

Continuing the pre-training of AraT5, we made the decision to utilise the same vocabulary model employed in the training of AraT5 by Nagoudi et al. (2022a). This vocabulary model was constructed using SentencePiece (Kudo et al., 2018), which encodes text as WordPiece tokens (Bostrom et al., 2020), comprising a total of 110,000 WordPieces. Consequently, our Vocab model is composed of 110,000 unique tokens. In cases where a word is not present in the vocabulary model, it will be identified as an out-of-vocabulary token.

7.4.4 Preprocessing and Cleaning of the Training Corpus

Training a language model requires using a large corpus. The training corpus should be preprocessed, cleaned and passed in a plain text format for training. In this training for FinaraT5, we will be using the corpus presented in section 5.3.1.
Data Acquisition: As stated in section 5.3.1, we collected several types of financial documents from different Middle Eastern markets: auditor reports, earning announcements, accounting documents, quarterly reports (Q1, Q2, Q3, Q4), annual reports and management board reports. A total of 30,000 PDF files were collected to form our source data. The total size of PDF files collected is around 25 GBs. However, we had several PDF files that were either corrupted or not suitable for PDF-to-text conversion. So, we had to make an additional selection of financial reports to be included in the training corpus.
PDF to Text Process: We started our conversion process by using the PDF to Text algorithm. We used the Sejda App PDF2Text API[5]. The significant constraint is the nature of the documents, which are scanned PDFs or contain old Arabic fonts

[4] The current models released by big corporations such as ChatGPT consists in several billion parameters: the size of models is increasing exponentially

[5] `https://www.sejda.com/extract-text-from-pdf`

or contain a lot of noise. In addition, the use of Arabic numerals and a lot of tabular data made converting to text files a challenging task. In total, we have around 8000 Arabic financial reports that we got from the PDF to Text process.
Arabic OCR solutions: The second solution is to opt for an Arabic Optical Character Recognition (OCR) solution to convert the files we did not manage to convert using the PDF2text algorithm. Arabic OCR is a technology that transforms PDF, printed or hand-written Arabic text into machine-readable textual data. The technology relies heavily on advanced machine learning to interpret Arabic characters' shapes, patterns, and structures (Alwaqfi et al., 2020).
Choice of OCR solution to be used: We compared different commercial OCR solutions to convert our PDF files into text files. Among these solutions, we highlight the following options:

- **GCP cognitive services**[6]: Google offers an NLP API that could be used to extract text.
- **AWS texract**[7]: This is a commercial cloud service from AWS that extracts the text from PDF files. However, it is limited to a maximum of 11 pages.
- **Adobe Acrobat DC Pro**[8] does not support the Arabic language for PDF to Text solutions.
- **Sejda app**[9] It is a SAAS solution to edit PDF documents(30+ PDF tools).

Conversion Process: We decided to go for a pro version of the Sejda app. As stated previously, We started using their PDF2Text algorithm to convert our PDF reports to plain text files. If the algorithm does not work, we use their Arabic OCR solution. The Arabic OCR inverts the order of words from left to right. So this has to be sorted out by inverting the order and making it Right To Left. Among the 30,000 collected reports, 24,000 were used in the process. We passed them through a pdf to text script through several batches. The PDF2text worked very well for several reports. The success rate was more than 40%. Some scanned docs were converted but generated ASCI code files, meaning the script cannot detect the content. Therefore we got around 8000 Arabic financial report text files from the PDF to Text process. For the others, we used the OCR tool of sejda[10]. On average, 10 PDF files take around 1 hour to be OCRed. The OCR operation took more than ten days, including the post-processing. We can not claim that the OCR solution of sejda is very efficient; however, it has an acceptable success rate regarding the poor quality of the report files. Finally, we did a manual check to verify that all the files had the minimum required Arabic structure for our pretraining process.

[6] https://console.cloud.google.com/marketplace/product/googlecloudvision/ocr-service-cpu
[7] https://aws.amazon.com/textract/
[8] https://www.adobe.com/uk/acrobat/acrobat-pro.html
[9] https://sejda.com
[10] https://www.sejda.com/ocr-pdf

We manually deleted all the badly converted files. We have in total less than 3000 successfully OCRed reports.

Challenges We describe the main challenges during the data construction and data conversion process from different aspects:

- **PDF2Text:** One of the common issues we observed from applying OCR on Arabic-written PDF files is getting repeated characters or additional spaces between the characters of one word (all the words are written with spaces) or concatenated words (not separated by spaces). This is reported to be a common issue for OCR in Arabic, especially if the data quality is not good enough.
- **Memory Management:** Producing such a large-scale corpus is very time-consuming; hence we divided the whole task into small tasks. It took around three months to construct the corpus, from web scraping until the last cleaned and pre-processed files were ready to be used in training.
- **OCR:** The low success rate for Arabic and especially a very long processing time given there was no possibility for parallel execution.

Financial Corpus Cleaning Once converted from PDF to text, we cleaned the text to be ready for the training. We used Farasa[11] for segmentation. We read files in chunks and apply our cleaning pipeline. This process started by removing all diacritics. We also removed HTML elements and their attributes. We removed all special characters and removed English alphabets and digits. We removed "tatweel" characters[12], which are used regularly in Arabic writing. We reduced repetitive characters to one and removed links and long words (longer than 15 chars). We used Tnkeeh tool (Alyafeai et al., 2020) to prepare our cleaning and preprocessing pipeline. The script 7.1 shows part of the code used to clean the corpus.

```

import tnkeeh as tn  # https://github.com/ARBML/tnkeeh

Path = <path folder containing files to clean>
Path_cleaned_version = <path folder cleaned files>

files = os.listdir(Path)

for file in files:
  Parallel_data_en_ar = os.path.join(Path_cleaned_version, 'Cleaned_' +
    file)
  abs_filepath = os.path.join(Path,file) # Path of a file to be cleaned
    .
```

[11] https://farasa.qcri.org/segmentation/

[12] tatweel character is used a lot in Arabic writing. It can be used in Ligatures, Poetry, Typography …

```
tn.clean_data(file_path = abs_filepath,
              save_path = Parallel_data_en_ar,
              segment=False,
              remove_special_chars=True,
              remove_english=True,
              normalize=False,
              remove_diacritics=True,
              excluded_chars=[],
              remove_tatweel=True,
              remove_html_elements=True,
              remove_links=True,
              remove_twitter_meta=True,
              remove_long_words=False,
              remove_repeated_chars=True,
              by_chunk=False,
              chunk_size=100000,
              normalize_dots=False,)
```

Listing 7.1: Preprocessing and cleaning script using tnkeeh library

In this sub-chapter, we described the process that enabled us to go from a raw PDF file Arabic Financial Report corpus presented previously in section 5.3.1 to a clean text corpus. The figure 7.2 summarises the whole process we described in this subsection to develop the pipeline used to prepare our training corpus.

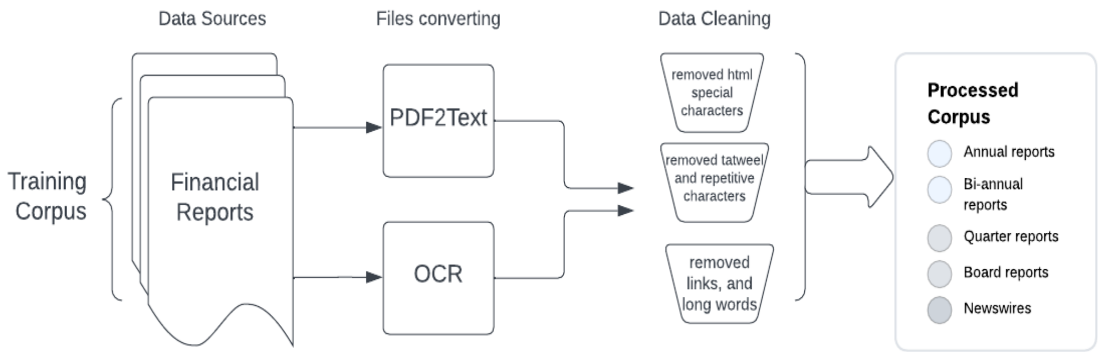


Figure 7.2: Financial Arabic Corpus Preprocessing Process: From raw PDF files to clean text files

7.4.5 Training Details

Pre-Training: we pretrain FinAraT5 on a TPU V-3.8 (with 8 cores) offered by Google cloud. we use a learning rate of 0.001. We used the Adam optimiser (Kingma et al., 2014). We fixed the batch size to 100,000 tokens. We set the maximum input and target sequence length to 512 sequences. We continued the training of the araT5 base for an additional 500,000 steps. We started from step 1 Million, where the arat5 was stopped. In total, we pre-train FinAraT5 for 1.5 M steps[13]. The pretraining took around 40 days without counting the time required for the araT5 base.

```

screen -L -Logfile finarat5_base_run.txt python3 -m t5.models.
   mesh_transformer_main
    --module_import="default_finaraT5_task"
    --tpu="tpu-finarat5-v3"
    --gcp_project="finarat5"
    --tpu_zone="us-central1-a"
    --model_dir="gs://finarat5/model/t5_mesh_checkpoints/"
    --gin_file="/home/user_name/text-to-text-transfer-transformer/t5/
   models/gin/objectives/span.gin"
    --gin_file="/home/user_name/text-to-text-transfer-transformer/t5/
   models/gin/models/t5.1.1.base.gin"
    --gin_file="/home/useer_name/text-to-text-transfer-transformer/t5/
   models/gin/dataset.gin"
    --gin_file="/home/user_name/text-to-text-transfer-transformer/t5/
   models/gin/learning_rate_schedules/rsqrt_no_ramp_down.gin"
    --gin_param="MIXTURE_NAME='unsupervised_default_finaraT5_task'"
    --gin_param="utils.run.train_steps = 1500000"
    --gin_param="utils.run.save_checkpoints_steps=6728"
    --gin_param="utils.run.batch_size=('tokens_per_batch',1048576)"
    --gin_param="utils.tpu_mesh_shape.tpu_topology = '2x2'" # V3.8
    --input_sequence_length=512
    --target_sequence_length=512
```

Listing 7.2: Script for Training FinAraT5

7.4.6 Pretraining Task

T5 (Raffel et al., 2020) was pretrained on a mixture of supervised and unsupervised tasks transforming each task into a text-to-text format. AraT5 (Nagoudi et al., 2022a) was pretrained using an unsupervised task. Therefore, we use the same pretraining strategy as AraT5, which is an unsupervised learning task trained on a raw plain text of qualitative financial data in Arabic. We cloned the architecture of T5 directly from the T5 GitHub repository[14]. We defined the task and performed the training using the t5 library[15], which enables us to perform the training using TensorFlow and get a Mesh TensorFlow Transformer.

Self-supervised learning has gained in popularity as it does not require labelled data. Technically, the unsupervised or self-supervised task consists of corrupting (masking) the input text and teaching the model to reconstruct it with the goal of

[13] We note that the English T5Base (Raffel et al., 2020) was trained only for 512K steps

[14] `https://github.com/google-research/text-to-text-transfer-transformer`

[15] `https://pypi.org/project/t5/`

NB: After this training, google released a new framework called T5x `https://github.com/google-research/t5x` that supports more efficient pretraining and use of cloud TPU. The t5 library that we used was very difficult to implement and needed very specific work of configuration to get the required results.

minimising a loss function (see figure 7.3). This does not require a labelled dataset. The self-supervised learning is recommended when we have a large textual corpus without labels. Therefore, with this training technique, our model is initialised (Pretrained) to be trained later on downstream financial tasks. But it always still needs fine-tuning. Another task that could be used to train language models is the next sentence prediction. But it was not used to train FinAraT5.

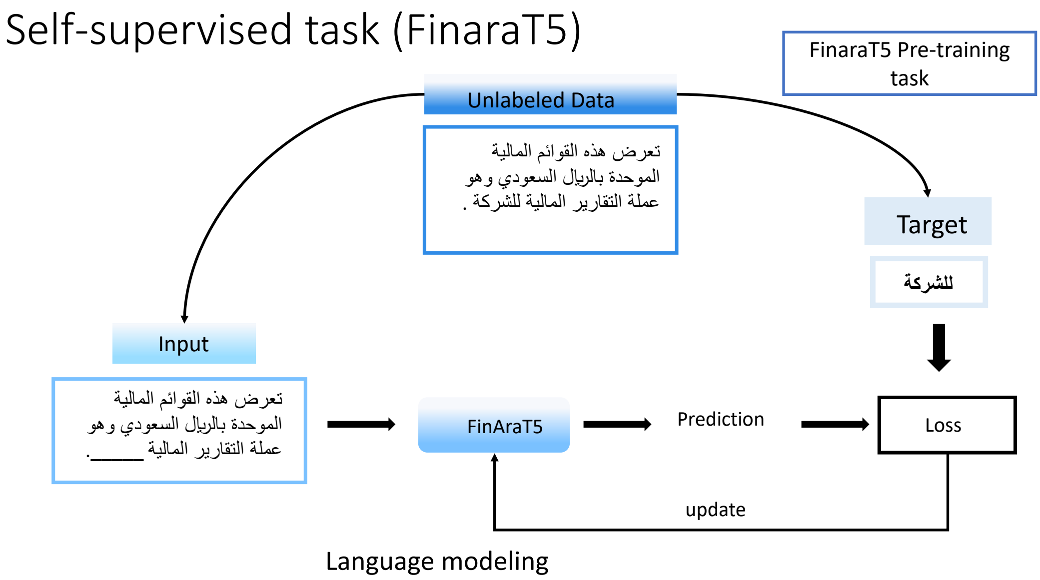


Figure 7.3: FinAraT5: A Self-supervised Pretraining Task

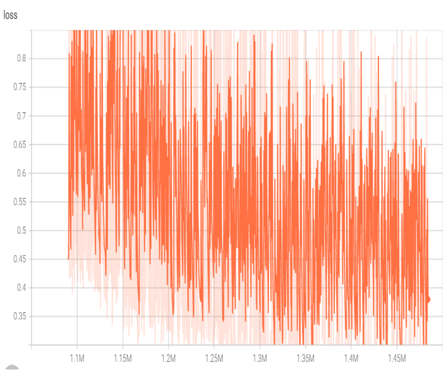
The script 7.2 shows the script example used to start the training. The use of 'screen' is to enable the process to run in the background to avoid a timing out on GCP and stop of the training process.

Tensorboard Plotting We used the Google Tensorboard tool[16] to plot some statistics and monitoring charts from the training process on GCP. Tensorboard enables monitoring of the loss and the learning rate. We report the loss, lr and global steps in figure 7.4.

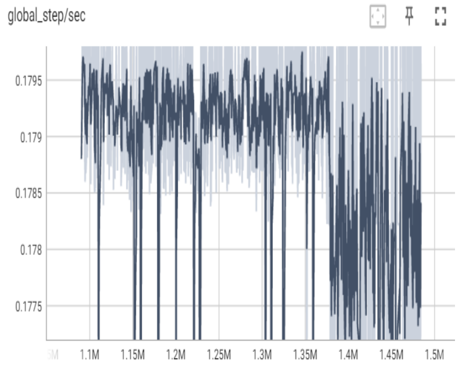
7.4.7 Experimental Setup

The pretraining was performed on the Google Cloud platform using TPU V3-8 accessed through the Google Cloud TPU API. The TPUs could be created using (ctpu[17]) Google tool, enabling us to provision, manage and delete TPUs. So after installing ctpu we create a TPU instance on a GCP project. We used the latest version of TPU and hosted it in Google's central US data centre. The training was

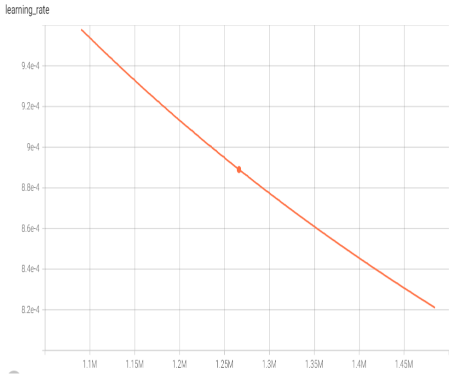
[16] https://www.tensorflow.org/tensorboard
[17] https://cloud.google.com/tpu/docs/ctpu-reference



(a) Loss Training FinAraT5



(b) Global Steps Training FinAraT5



(c) Learning Rate Training FinAraT5

Figure 7.4: Tensorboard Plotting of FinAraT5 training

done with the help of some libraries such as mesh-tensorflow, sentencepiece, t5 , tensorflow-text, tfds-nightly, torch, transformers and seqio. And the storage was done through a Google Bucket.

7.4.8 Model Visualisation (Bertviz)

In this section, we will visualise the inner parts of attention of our pretrained model by using some Transformer visualisation tools. This visualisation is essential to interpret the learned representations and to understand the information encoded by self-attention heads in the Transformer. We used BertViz[18], which is a transformer visualisation tool created by Vig (2019). It enables in an efficient way to visualise the attention layers of the Transformer model (Pretrained or finetuned). It was built on top of the work of Tensor2Tensor's visualisation tool[19], and it allows for more detailed multi-scale examination of the model's inner layers and heads. BertViz provides a unified interface that displays the attention heads and offers users a broad overview as well as a detailed view of how individual neurons interact to form the attention

[18] Bertviz: `https://github.com/jessevig/bertvizBertviz`
[19] `https://github.com/tensorflow/tensor2tensor`

weights (Vig, 2019). BertViz supports three views: a head view, a model view, and a neuron view. The most significant advantage of BertViz is that it works with most Hugging Face-hosted models through the Python Application Programming Interface (API), and we should note that in BertViz, we start from 0 for indexing.

Attention head view: The head view visualises attention for one or more attention heads in the same layer. It is based on the Tensor2Tensor visualisation tool.

Model View: The model view shows a bird's-eye view of attention across all layers and heads. It allows us to have a general perspective of the focus of all heads and layers. The self-attention heads are displayed in a table, with rows and columns representing the different layers and heads, respectively. Each head includes an outline of the attention model by clicking on it.

Figure 7.5a, Figure 7.5b and Figure 7.6 show an internal visualisation of attention across all of the model's layers and heads for a particular input and output. The three figures represent respectively, the encoder, decoder and a cross representation(Encoder-Decoder). The attention heads are organised in the format of tables, where columns show heads and rows indicates layers. We used these two sentences for the input and output, respectively in order to test our models.

الذهب يصعد مع ارتفاع التضخم في أميركا

Translation: Gold price rises with the rise in inflation in America.

الذهب يرتفع مع ارتفاع التضخم في أميركا

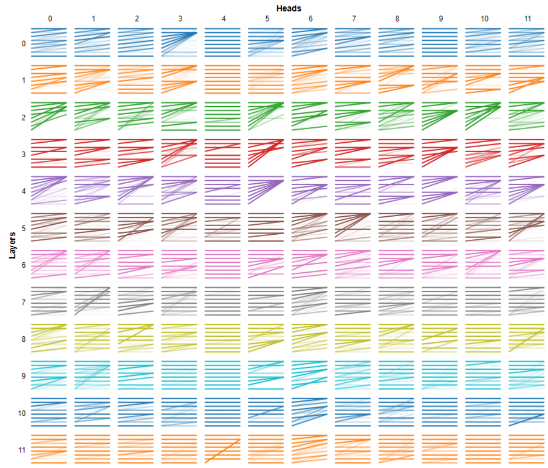
Translation: Gold price increases with rising inflation in America.

The model view (illustrated in Figures 7.5 and 7.6) enables us to quickly browse the attention heads across all layers and to see how attention patterns evolve throughout the model. This view is particular to a specific input and output mentioned previously. Plotting this view helps us quickly observe many patterns, such as next-token or previous-token attention patterns. Also, it helps understand how attention evolves through layers.

The self-attention mechanism is crucial, especially in Arabic, because it helps understand the meaning of the words inside a sentence. In Arabic, some words could have several meanings depending on the context or where they are used in the sentence. This mechanism becomes more complicated since we do not use diacritics. These graphs show how every word is connected to other words, whether in the encoder or decoder or a cross-visualisation (encoder, decoder). Self-attention will mean that every word (or token) in an input sentence will have access (or see) to other words in the same sequence with certain weights for every word. It means every word on the left side sees words on the right side. The Weight is high when the colour is dark, and it is low when the colour is light. Higher weights mean high similarity between the two tokens in the sentence context.

The left encoder and the right decoder of the FinAraT5 transformer are linked by means of cross-attention, which aids each decoder layer in being aware of the ultimate encoder layer. This encourages models to produce output that is strongly

(a) The model view of the FinArat5 language model encoder



(b) The model view of the FinArat5 language model decoder

Figure 7.5: The model view of the encoder and decoder of finarat5 language model

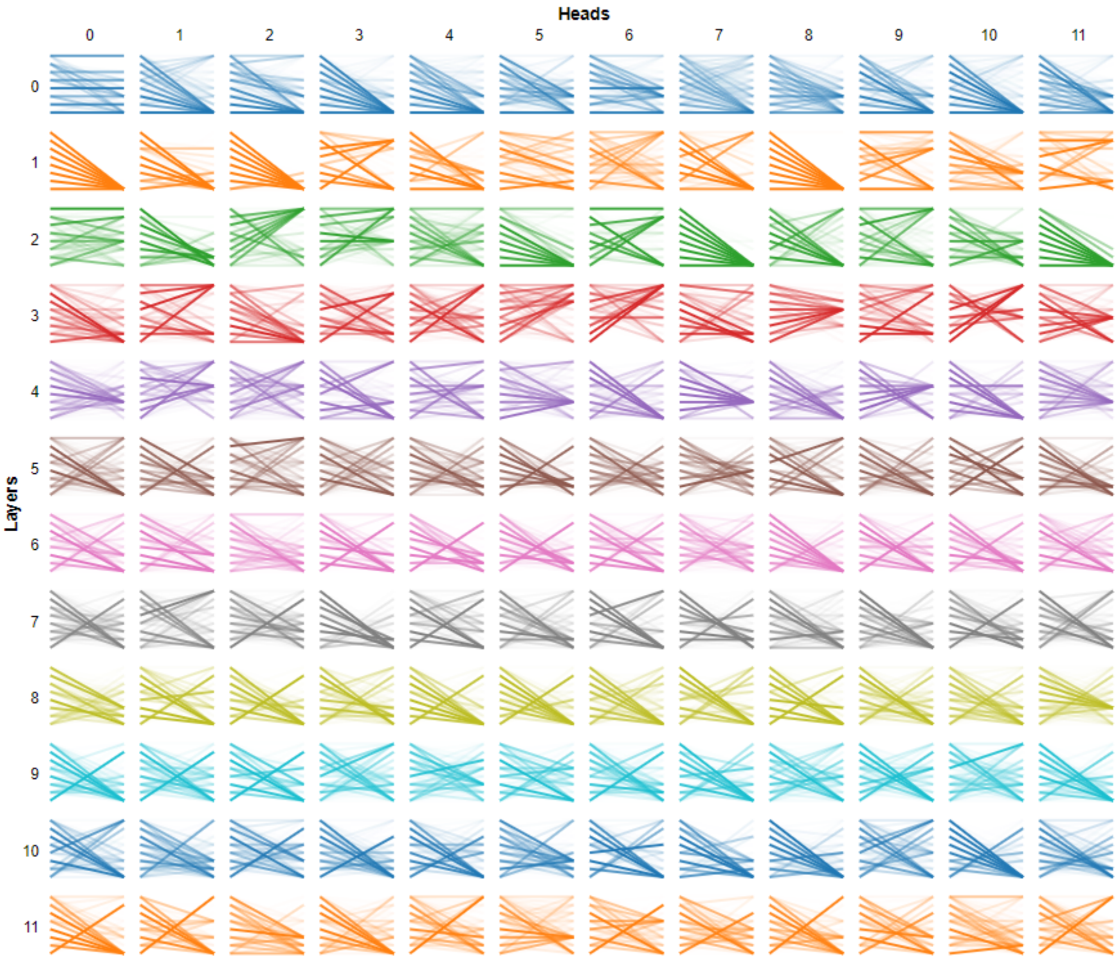
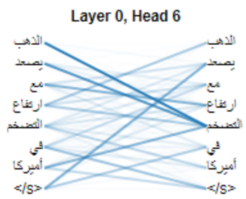
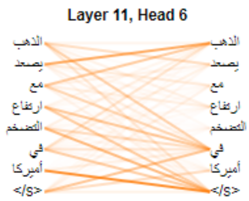


Figure 7.6: A cross model view of the FinArat5 language model

(a) L6 H6

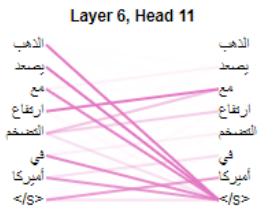


(b) L0 H6



(c) L11 H6

(d) L6 H0



(e) L6 H11

(f) L6 H6

Figure 7.7: Zoom in on different layers and heads from Finarat5 encoder Bertviz visualisation

(a) L3 H2

(b) L8 H5

(c) L5 H5

Figure 7.8: Zoom in on different layers and heads from base decoder Bertviz visualisation

associated/coherent with the original input. A text2text model achieves this by utilising this process: Input tokens → embeddings → encoder → decoder → output tokens.

If we zoom in on the encoder visualisation reported in figure 7.5a, we will have the figure 7.7 where we include figures of some heads and layers. This shows us the attention mechanism (How every word is linked to other words of a given sequence). If we also zoom in on the decoder visualisation reported in figure 7.5b, we will have the figure 7.8 which also shows some heads and layers from the Bertviz visualisation of the decoder.

7.5 Experiments

7.5.1 Arabic Financial Text Summarisation - Headline Generation

The process of generating headlines automatically is a subset of document summarisation. The task of headline generation was addressed several times in past summarisation challenges, such as the Document Understanding Conferences (DUC) for the 2002 [20], 2003 [21] and 2004 [22] editions. This task could be solved using three techniques: rule-based, statistic-based, and summarisation-based.
We trained mT5 small, base, and large. We were unable to train the mT5 Xlarge due to memory limitations. We also trained arat5 small, arat5 base, finaraT5 base and bert2bert base. For bert2bert, we followed the methodology proposed by Rothe et al. (2020). So we created a sequence-to-sequence model whose encoder and decoder parameters are multilingual uncased Bert base model[23]. We will oblige the multilingual Bert model to work as an encoder and a decoder to generate the summary. All models were trained for 22 epochs, and we used a learning rate that warmed up to 5e-5 . We did not allow tokenisation parallelism during the training. In addition, we used early stopping and took the best checkpoint on the validation split. The models used here are:

- **araT5** (small, base) (Nagoudi et al., 2022a): It is an Arabic T5 model.
- **mT5** (small, base, large) (Xue et al., 2021): mT5 is multilingual model pretrained on the mC4 corpus, covering 101 languages.
- **bert2bert (B2B)**: We followed the methodology proposed by Rothe et al. (2020). So we created a sequence-to-sequence model whose encoder and decoder parameters are multilingual uncased Bert base model[24]. We will force the mBert model to work as a decoder to generate the summary. This is the most challenging model to train because we had to do the process manually without using the trainer API from huggingface. The training and prediction process is different from the sequence-to-sequence pretrained models.

LEAD-1 baseline is included, a competitive extractive baseline for news summarisation by extracting the first sentence (Gallina et al., 2020). The structure of news articles usually follows an “Inverted Pyramid” where essential information comes first (Dai et al., 2021). There is a paradigm called Lead Bias that motivates the first three sentences of a news article to be often used as a baseline for summarisation (Yang et al., 2020b). However, in this case of Arabic abstractive summarisation,

[20] https://www-nlpir.nist.gov/projects/duc/guidelines/2002.html
[21] https://www-nlpir.nist.gov/projects/duc/guidelines/2003.html
[22] https://www-nlpir.nist.gov/projects/duc/guidelines/2004.html
[23] https://huggingface.co/bert-base-multilingual-uncased
[24] https://huggingface.co/bert-base-multilingual-uncased

lead-1 is a weak baseline for two main reasons: First because the gold standard is highly abstractive and it introduces new n-grams, and the second reason is the fact that the average length of lead-1 is very longer compared to the gold standard and system generated summaries. So, in this case, we do not expect good performance form the lead-1. Globally the idea here is to compare different text-to-text models (base, large, small) and use two baselines bert2bert and lead. The training was done on a Conda[25] environment 7.3 on the High-End Computing Cluster of Lancaster University [26].

```
channels:
  - pytorch
  - defaults
  - conda-forge
dependencies:
  - python=3.8
  - pip
  - pytorch=1.7.0
  - cudatoolkit=10.2
  - conda-forge::git
  - conda-forge::git-lfs
```

Listing 7.3: Conda Environment Financial Summarisation Finetuning

7.5.2 Evaluation

Evaluating text generation in general and the headline generation task, in particular, is very tricky and subjective, especially in the case of short headlines and a very rich language such as Arabic (see section 7.1). Two sentences in Arabic can have very similar embeddings but different meanings because the Arabic language is a rich language where one word could have different meanings. In addition, a comma or a semicolon could change the meaning. Automatic metrics can not always flag these semantic specificities of Arabic. That is why evaluating Arabic-generated text is a big open question. Therefore, we should add a qualitative study and even manual testing of the models we want to test in order to see in practice what this model can do in terms of high grammatical and semantical Arabic mastery.

In this section, we will explore different evaluation methods. We will start by using a quantitative metric based on trained embeddings such as frugalscore. Then, we will check the clarity and non-redundancy of the generated text. To do that, we will calculate the percentage of repeated words among the top n words (we need to make sure that the model is not redundant and does not create hallucinations). After that, we will perform an evaluation based on the readability score of the generated text, which is a crucial step. We need to check whether the generated text is readable, which is more important than just trying to match a gold standard summary using quantitative metrics. Afterwards, we will present a small human

[25] https://docs.conda.io/en/latest/

[26] https://www.lancaster.ac.uk/iss/info/IThandouts/hec/HEC-flyer.pdf

evaluation experiment which enables us to show a different point of view or a technique of evaluation. Human evaluation could be the benchmark against quantitative evaluation, which is not accurate, as stated in section 2.7 and sometimes non-statistically significant, as stated in section 6.8. Finally, we will perform a manual evaluation of the abstractiveness of the models by giving some statistics about the percentage of new unigrams and bigrams introduced by the models and we will end by analysing different grammatical aspects of text generated by the finarat5 model.

For the evaluation metrics, motivated by Kamal Eddine et al. (2022), we decided not to go for Rouge (Lin, 2004) and meteor (Banerjee et al., 2005) (string-based metrics) although they gave good results in this case of abstractive summarisation. This decision is motivated by the fact that rouge and meteor have serious limitations for evaluating abstractive summaries. Rouge variants are based on n-gram-form matching and have no sense of semantic similarity (Kamal Eddine et al., 2022; Kamal Eddine et al., 2021). Furthermore, string-based metrics rely mainly on lexical overlap, whereas an abstractive summary could express the same meaning as a gold standard without any lexical overlap. This assumption may be real in a special language like Arabic, where we can express the same meaning using different n-grams. In addition, Rouge does not take into account coherence, fluency and grammar (see section 2.7.2). Furthermore, rouge variants were designed to be used with multiple reference summaries per input rather than datasets with one gold standard for every input text. That is why we will go for alternative ways to evaluate to have a more objective point of view.

Instead, we can, for example, use frugal score (Kamal Eddine et al., 2022) which is a cheap implementation of BERTScore (Zhang* et al., 2020) and Moverscore (Zhao et al., 2019) running faster with minimum GPU requirements (see table 7.1 for the results on the headline generation task). Although Frugalscore does not generate significant difference between model results, it is cheap and efficient to implement. Frugalscore 1 (FS1) uses the Bertscore methodology with Bert as a language model, whereas Frugalscore2 (FS2) uses the Bertscore methodology with RoBERTa as a language model. Frugalscore 3 (FS3) uses the Bertscore methodology with DeBERTa as a language model. Frugalscore 4 (FS4) uses the moverscore methodology with Bert-tiny as a student. Frugalscore 5 (FS5) uses the moverscore methodology with Bert-small as a student. Finarat5 shows competitive results compared to multilingual versions of mT5, especially with Base and Small models. It outperformed all the small and base models. This confirms the importance of pre-training monolingual models. Finally, all T5-based models outperform bert2bert and Lead (the two benchmarks) by a significant margin. We should mention here that we are comparing finarat5 and arat5 with mT5 model variants (Xue et al., 2021), which are much bigger than regular T5 architectures due to a larger vocabulary and embedding matrix. However, this quantitative evaluation is insufficient and may be inaccurate, as stated earlier in this section. Evaluating using quantitative metrics is not sufficient because there is a high risk of hallucination, repetition and generated fake information that could easily match the embeddings of the gold standard but does not reflect the real meaning that we want. Furthermore, in the case of reinforcement learning, we can set a reward function (E.g. Frugal score or rouge-2), and we can train a model to maximise this reward function without focusing on the quality of generated text (This is the case of RL systems developed in chapter 5 where we focused on maximising rouge-2 rather than on generating readable and coherent text). For all these reasons, we will explore other evaluation methodologies and criteria.

	title generation				
	Frugal 1	**Frugal 2**	**Frugal 3**	**Frugal 4**	**Frugal 5**
lead	77.38	93.69	86.30	85.10	45.75
mT5 small	83.03	95.02	89.30	91.50	66.23
araT5 small	83.23	95.09	89.40	91.65	66.94
bert2bert	82.83	95.01	89.20	91.26	63.74
mT5 base	83.11	95.03	89.30	91.46	66.19
araT5 base	83.49	95.14	89.50	92.04	68.25
finaraT5 base	83.49	*95.14*	*89.52*	*92.04*	*68.30*
mT5 large	83.49	95.11	89.45	92.04	68.40

Table 7.1: Frugal score results.

Another criterion for a good summary is clarity and non-redundancy, as stated in section 2.7.1. So, we calculated the percentage of repetition and the average length of the system-generated summary (nine tokens on average). The repetition rate is the rate of summaries including at least a one word repetition from the most frequent n words from the corpus. Results are detailed in Table 7.2 where we report the sequence length generated by models and the rate of repetition in the summary among the most common 200, 300, 400 and 500 words in the dataset. For repetitions, the less redundant model closest to the ground truth is mT5 large. The use of auto-generative models on abstractive datasets increases the risk of repetition. Our model finarat5 shows less repetition on this summarisation dataset than other models. This is a good sign of the quality and novelty of the generated text.

We can see that the larger the model is (in number of parameters) the less repetition we have. The gold standard summary stands as the best model in terms of repetition of words from the most frequent words in the corpus. This is justified by the fact that it is an unsupervised model generated by humans. Humans have access to a larger vocabulary, especially in the case of native human speakers. However, models can only use vocabulary from the seen set of words. So, they may have recourse to repetition. Bert2Bert is the only model redundant with 15.76% of repetitions, which may be justified by the architecture of this model (being a non-native seq2seq model). Further manual checking of the bert2bert model shows that it sometimes causes some hallucinations or generates irrelevant words. In addition, this model generated more tokens on average. This is coherent with previous results.

	Length	**rep 500 %**	**rep 400 %**	**rep 300 %**	**rep 200 %**
Gold	9.04	0.4	0.52	0.52	0.72
mT5_small	9.27	4.36	4.44	4.64	5.32
araT5_small	9.28	5.32	5.64	6.04	6.56
bert2bert	10.03	15.08	15.76	16.48	17.36
mT5_base	9.05	2.56	2.64	2.76	3.24
araT5_base	9.08	3.36	3.64	3.76	4.2
finarat5_base	9.05	3.24	3.48	3.84	4.48
mT5_large	8.92	1.28	1.2	1.36	1.88

Table 7.2: Summary length and repetition statistics

This evaluation is still limited and needs more advanced analysis. As stated previously, in Arabic, two sentences written with different syntax and vocabulary could have the same meaning and similar embeddings. This risk becomes higher in the case of abstractive summarisation, especially when we are generating a headline and not a long summary. However, one sentence could be stylistically more formal and correct, respecting the standards of the Arabic language. That is why we should find a more advanced way of evaluation. We should focus on whether the generated text is readable and coherent rather than it just matches a gold standard. As stated in section 2.7.1, readability is a crucial criterion to measure the quality of a summary. Therefore, we will check the readability of the generated text, and we will do a human evaluation. Then, we will analyse some grammatical aspects of the generated content.

So, we will explore an important measure to evaluate the model, which is the readability of generated text. We will use the readability score as an objective evaluation criterion. Quantitatively, a model could have a decent performance, but it may generate non-coherent text that matches the semantics or embeddings of the gold standard. Here, we would ensure that the model generates readable text, which is an important feature. So, the idea here is to calculate the correlation between the Arabic readability score of the generated text and the readability score of the gold standard summary. At least we are sure that the human-written headline is readable and follows the grammar and syntactical aspects of standard Arabic. Combining this evaluation with previous results, we can have a clearer image of how to efficiently evaluate a model from different sides. We will use the OSMAN readability metric, which is a free, open-source metric and tool to measure the readability of Arabic text with or without diacritics (El-Haj et al., 2016). Additionally, our approach incorporates a new factor called "Faseeh" which considers features of script usually excluded in informal Arabic writing. The Osman readability score is expressed as follows 7.2

$$Osman = 200.791 - 1.015 \times \left(\frac{A}{B}\right) - 24.181 \times \left(\frac{C + D + G + H}{A}\right) \tag{7.2}$$

‘A’ number of words ‘B’ number of sentences ‘C’ number of hard words (>5 letters) ‘D’ number of syllables ‘G’ number of complex words (>4 syllables) ‘H’ number of “Faseeh” words (complex word containing (‘ء’, ‘ئ’, ‘ؤ’, ‘ذ’, ‘ظ’) or ending with (‘وَا’, ‘وْنَ’).

Using OSMAN readability score, we can examine how well Arabic readability scores align with this objective measure of quality. To do this, for each generated title, we compute the correlation between the readability score of the automated summary and that of the gold standard version. In the first step, we calculated the Osman readability metric for all the system summaries generated by the different models and the gold standard summary. Then, the table 7.3 includes the correlation score between the readability score of the human gold standard and the readability scores of different systems. We report Pearson r (Kowalski, 1972; Zabell et al., 2008; Bishara et al., 2017; Wikipedia contributors, 2010), Spearman ρ (Kokoska et al., 1999; Beech, 1962) and Kendall τ (Kendall, 1938; “Front Matter” 1945; Yen, 1968) (Definitions included in appendix C.4). Finarat5 has the highest correlation with the gold standard on readability metric in two correlation measures (Pearson r (0.5110) and Kendall τ (0.3610)), while the arat5 has the highest correlation using Spearman ρ . This evaluation proves the high importance of monolingual models, especially in the Arabic language, compared to multilingual counterparts. Although mT5 versions are strong models able to generate Arabic text, monolingual-trained models have the competitive advantage of generating very readable text close to human-generated text. This encourages training more monolingual field-specific models. Furthermore, this proves that multilingual models could have strong quantitative scores but still generate less readable text than monolingual models (This is a crucial criterion in such a rich language such as Arabic).

System	**pearson r**	**spearman ρ**	**kendall τ**
finarat5_base	**0.5110**	0.5089	**0.3610**
araT5_base	0.5071	**0.5094**	0.3607
araT5_small	0.4684	0.4691	0.3282
mT5_small	0.4510	0.4468	0.3121
mT5_base	0.4341	0.4301	0.3010
mT5_large	0.4759	0.4699	0.3299
bert2bert	0.3883	0.3760	0.2592

Table 7.3: Correlation between the Osman readability metric of different systems and the gold standard.

To conclude this section, we should mention that the major drawback of quantitative methods is that we are always comparing models to the gold standard and aim to match a gold standard. However, practically, we do not train a model to replicate an existing headline. Rather, we aim to beat it. In addition, in this case, we have only one gold standard, unlike the FNS dataset used in chapter 6, which makes the quantitative evaluation non-objective, especially since we can not affirm that the current title of the news article is the best although it is written by a human journalist. Furthermore, in some cases, a highly abstractive generated system summary can be excellent but does not match the gold standard. It may overperform the gold standard summary, but quantitative evaluation does not show this. This can be justified by the rich vocabulary of the Arabic language and the ability to express the same meaning using different words and structures. For example, the word lion has more than 400 names in Arabic. There are 255 for the camel, 300 for the sword, 170 for water, 70 for rain, and each of these words or synonyms have a particular usage[27]. Hence, the best thing we can dream of is implementing a human-in-the-loop evaluation system. This is where we can have the more significant evaluation in the case of the highly abstractive dataset. In the next section, we will perform a small human evaluation task with limited resources. However, it may help clarify the process more since human assessment is still the standard benchmark for text summarisation, as stated in section 2.7.3. It will enable to compare the system summaries against the gold standard, which is exactly the scientific aim of summarisation (generating readable content that beats the gold standard), rather than aiming to maximise the matching between the gold standard and the system summary.

7.5.3 Human Evaluation

In this section, we will describe the human evaluation task that we performed on the output of different models trained on the task of financial Arabic news summarisation. Apart from using automatic evaluation, which can be uncertain when used alone to judge summary informativeness (Schluter, 2017), we also conducted a human evaluation experiment. We should mention that, if performed on a large sample, human evaluation could be more accurate than quantitative evaluation (see section 2.7.3). Following Narayan et al. (2018c) and Kamal Eddine et al. (2021), we used Best-Worst Scaling (Louviere et al., 2015; Louviere et al., 1983). Kiritchenko et al. (2017) showed that compared to rating scales, Best-Worst Scaling is both faster and more accurate.

The process was as follows: two summaries from two different supervised systems and their input document are presented to a human annotator (native Arabic speaker) who should decide which one is better. We asked evaluators to base their judgments on **accuracy** (does the summary contain accurate facts? Some models may invent some numbers or mention wrong dates), **fluency and readability** (is the summary written in well-formed Arabic respecting the grammatical specificities of this language?), **informativeness** (is important information captured?), **Typos** (focuses on if the generated summary has some typos) and **Succinctness** (measures whether the summary is concise and does not describe too many details). These criteria are motivated by our task being an abstractive summarisation task. We

[27]Source: `https://www.fluentarabic.net/beautiful-arabic-language/`

want to check if the model is generating readable content. The evaluation criteria such as informativeness, fluency and succinctness were used in prior studies such as (Huang et al., 2020). The final score of a model is given as the percentage of time its summary was chosen as best minus the percentage of time it was selected as worst. The scores range from -1 (the lowest) to 1 (the highest). The more the score is, the better the model is. We can have negative results. The negative score means that its summaries were judged to be worse more often than not. (Check appendix C.3 for additional information about the human evaluation task)

The Arabic Native speaker evaluators were mainly from a computer science and STEM background. They had a minimum understanding of economic and financial news articles and especially had a deep understanding of nuances in the Arabic language. We included five systems in the study: gold summary, finarat5, arat5, mt5 and bert2bert. Adding another system will exponentially increase the number of evaluation tasks. We randomly sampled 45 documents and compared all possible combinations of two out of five systems for each document. We generated all possible pairs of FinAraT5, mT5, bert2bert, araT5 and gold standard summaries for each document, resulting in 450 pairs. Each pair is assigned to three different annotators. Hence, this resulted in a total of 1400 evaluation tasks.

Explanation: number of tasks We include five summarisation systems(FinAraT5, mT5, bert2bert, araT5 and gold) over 45 financial news. For every news article, we will get 4+3+2+1 = 10 pairs. Then, we will have 45*10 = 450 evaluation pairs for the total. We aim for three reviews per pair. We will have 1400 evaluation tasks.

The number of times each model is involved in an evaluation task: Every system will enter a comparison with four systems for every news article, resulting in 45*4 = 180 tasks. Since we aim for three different evaluations for every task or pair, the total number of tasks involving each system would be 180 * 3 = 540.

That is why the final score is $BWS = \frac{\#selected\_best - \#selected\_worst}{540}$

Results and analysis: Table 7.4 shows Human evaluation scores using Best-Worst Scaling

System	**Score**
gold	**0.3556**
araT5	0.1370
finarat5	*0.2963*
mT5	0.0333
bert2bert	-0.8222

Table 7.4: Human evaluation scores using Best-Worst Scaling.

The gold standard is ranked best by human evaluators with a 0.35 score. However, in several cases (more than 170), a system-generated summary outranked the gold summary. This motivates the hypothesis that automatically generated text could outperform human-written text. Furthermore, the three text-to-text models have positive results, which means that the number of times they were judged best is more than the number of times they were judged worst, which is a good sign for system-generated summaries. The very negative score of bert2bert means the system summaries were judged worse in most cases, which is not surprising since bert2bert does have some hallucinations. We may think it is a good summary if we do not read the article. However, in reality, it is sometimes inventing false stories. In addition, the two Arabic models (finarat5 and arat5) have competitive results with the gold truth ones. We hypothesise that readability and coherence are the most important criteria since the results are similar to the readability scores we got in the previous section. It is much easier for a human evaluator to check the readability and grammatical aspects of a headline rather than check the accuracy and succinctness.

We should also mention that there is not always an agreement between the evaluators for every task, which confirms the subjectiveness of the evaluation process. One major feedback from the reviewer is that it is often very difficult to choose the best summary, confirming the difficulty of human evaluation. Choosing a better headline is very subjective. One person would prefer simple and short headlines; another would select more complicated ones using advanced grammatical and stylistic structures. But the human evaluation is still an interesting exercise. In fact, in some cases, a headline seems fine, but it includes a typo or sometimes the writing of a character differs if it is in the beginning, end or middle of the sentence in Arabic. These specificities of the Arabic language can not be flagged by quantitative evaluation. Furthermore, we should mention that the results are not statistically significant since the sample is so small, and we cannot go for further samples due to cost reasons, so we can not generalise. Still, it gives an idea of how we can perform human evaluation. Future work in case of a decent budget would be to perform a large-scale evaluation with master evaluators on a crowdsourcing website such as mechanical turk[28]

7.5.4 Example Summaries

Table 7.5 show an example of a news article, the gold summary and different model-generated summaries. We include an English translation. In addition, appendix C.2 presents some more examples of the system-generated summaries of various models we trained.

[28] `https://www.mturk.com/`

	Document	تراجعت أسعار الذهب اليوم الخميس مع صعود الدولار بعدما رسم مجلس الاحتياطي الاتحادي (البنك المركزي الأميركي) صورة إيجابية للتعافي الاقتصادي لكنه أحجم عن تقديم مؤشرات ملموسة بشأن المزيد من التحفيز.، وبحلول الساعة ٠٥٢٤ بتوقيت غرينتش، نزل الذهب في المعاملات الفورية ٨٠٠ % إلى ٧٠٠١٩٤٣ دولار للأوقية (الأونصة) فيما تراجعت العقود الأميركية الآجلة للذهب ١ % إلى ٥٠٠١٩٥٠ دولار.، وارتفع مؤشر الدولار إلى أعلى مستوى في أكثر من أسبوع مقابل بقية العملات بعدما ألمح الاحتياطي الاتحادي أمس الأربعاء إلى أنه يتوقع تسارع تعافي الاقتصاد الأميركي من جائحة كورونا وتراجع البطالة بوتيرة أسرع مما توقع البنك المركزي في يونيو حزيران.، وتلقى الذهب بعض الدعم من تعهد البنك المركزي بإبقاء أسعار الفائدة قرب مستويات الصفر مئوية حتى يمضي التضخم على مسار آتجاوز باعتدالْ لمستوى ٢ % المستهدف آلبعض الوقتْ.، وبالنسبة للمعادن النفيسة الأخرى، انخفضت الفضة ٥٠١ % إلى ٨٢٠٢٦ دولار للأوقية وتراجع البلاتين ٣٠٢ % إلى ١٠٠٩٤٦ دولار للأوقية. ونزل البلاديوم ٣٠١ % إى ٤٩٠٢٣٦٧ دولار للأوقية.
Title	Gold	بورصةالذهب يتراجع بعد ارتفاع الدولار وسط توقعات اقتصادية قوية The gold stock market is falling after the rise of the dollar amid strong economic expectations
	mT5 small	االذهب يتراجع مع صعود الدولار Gold is falling with the rise of the dollar
	araT5 small	الذهب يتراجع مع صعود الدولار مع صعود الدولار Gold is falling with the rise of the dollar With the rise of the dollar
	bert2bert	الذهب يهبط مع صعود الدولار وصعوده بعد رسم البنك المركزي الاميركي Gold falls with the rise of the dollar and its rise after drawing the US Central Bank
	mT5 base	الذهب يتراجع مع صعود الدولار بعد رسم بيانات إيجابية للتعافي الاقتصادي Gold is falling with the rise of the dollar after drawing positive data for the economic recovery
	araT5 base	الذهب يتراجع مع صعود الدولار بعد تصريحات الفيدرالي Gold is falling with the rise of the dollar after the Fed's statements
	finaraT5 base	الذهب يتراجع مع صعود الدولار بعد تصريحات المركزي الأميركي Gold is falling with the rise of the dollar, after the US Central Bank's statements
	mT5 large	الذهب يتراجع مع صعود الدولار بعد رسم صورة إيجابية للتعافي Gold is falling with the rise of the dollar, after painting a positive picture of recovery

Table 7.5: Different generated abstractive headlines showing the output of the various systems we trained.This article belongs to the "financial markets" category.We provide English translations to provide context for the general readers. The translation may not reflect the exact meaning.

7.5.5 Discussion and Analysis

Multilingual vs. Monolingual Models The empirical results show the better performance of dedicated monolingual language models compared to multilingual models (multilingual T5 versions: 110 languages) of the same size. The FinAraT5 model benefits from the previously pretrained araT5 on a large Arabic corpus. In addition, it specialises in the financial context by being trained on a large financial narrative corpus.

Abstractiveness: The abstractiveness can be defined as the ability to generate new words that do not exist in the original text as shown in table 7.6. We can see the superiority of Arabic monolingual models for the percentage of novel unigrams, bigrams and trigrams. We can deduce that FinaraT5 and araT5 base are more abstractive than other models, especially multilingual T5. Bert2Bert is an exception since it generates some random words and some kind of hallucination. That is why it is generating more new words. This may be justified by the fact that it is not a native encoder-decoder model.

Model	**% of new n-grams**		
	Unigrams	**Bigrams**	**Trigrams**
Gold	37.1	73.1	88.8
bert2bert	34.2	77.3	95.4
mT5 small	22.1	52.8	71.1
araT5 small	27.5	62.2	80.4
mT5 base	23.7	54.2	72.6
araT5 base	28.3	63.9	82.4
FinAraT5 base	28.8	64.5	82.6
mT5 large	26.3	60.8	79.5

Table 7.6: Percentage of novel unigrams, bigrams and trigrams in the generated headlines.

Grammatical aspects: In this section, we will not discuss the accuracy of the content. We will only judge the grammatical aspect. We manually analysed our text-to-text models' ability to generate good-quality financial context in modern standard Arabic text. The generated text is syntactically correct, and the spelling is correct, too. It is also in line with the general topic of the corpus. The Finarat5 model introduces advanced grammatical Arabic structures, such as using question marks, exclamations, and oratorical questions. In addition, we see good use of commas, which is crucial in Arabic, enabling emphasis on some words. Finally, we can see that different versions of Arabic T5 generate content that has approximately the same meaning using different structures. In conclusion, our models can generate syntactically correct summaries in Arabic, which is a very challenging language with many words having different meanings depending on context. Also, the Arabic language uses a lot of metaphorical techniques.

finarat5:

خبراء: ترقية سوق السعودية تعزز الثقة بأسواق المال

Translation: Experts: Upgrading the Saudi market enhances confidence in financial markets.

gold:

ترقية السوق السعودي إلى آناشئةْ يحفز الاستثمار الأجنبي

Translation: Upgrading the Saudi market to "Nachiaa" stimulates foreign investment.

This is the first example where we show the grammatical and structure use case behind using custom models. In fact, introducing a comma is a well-known grammatical structure used in Arabic to emphasise a specific word or a whole idea. In this specific case our model is able to understand the author of the claim and put it at the beginning of the generated title, then add colon before announcing the news. This is one of the highest levels in the classical Arabic language.

Further examples show that finarat5 generates some titles using question marks. Question marks are a mode of writing widely used to create suspense for readers and encourage them to click on the link; in this case, the gold standard gives you the news directly, whereas the finarat5 title motivates you to read the whole article. This is another example proving the potential of generative models and their usefulness in automating some parts of the news production industry. With custom language models, we can choose the tone and the grammatical structure.

finarat5:

عملات الأسواق الناشئة تتكبد خسائر عنيفة.. ما هي؟

Translation: Emerging market currencies incur heavy losses.. What are they?

gold:

هوس خسائر الليرة التركية يعصف بعملات الأسواق الناشئة

Translation: The obsession with the losses of the Turkish lira is ravaging emerging market currencies.

In the same context of introducing question marks, we present the following example where we have an innovative use of oratory questions. The two titles give the same idea; however, finarat5 adds this oratory question that adds an additional level of grammatical complexity and innovation for Arabic content. The question does not require an answer. It is used to emphasize the bad financial situation of the company.

finarat5:

دينهامز تواجه خطر الإفلاس في بريطانيا.. فهل انتهت جهود إنقاذها؟

Translation: Debenhams faces the threat of bankruptcy in Britain..Has the effort to save it ended?

gold:

أشهر محل أزياء بريطاني مهدد بالإفلاس بعد ٢٤٢ عاماً من العمل

Translation: The most famous British fashion store is threatened with bankruptcy after 242 years of operation.

Another advanced use case is where our model detects the date written in letters in the body of the article, and transforms it into numbers, which is more convenient in a headline, then the model puts the source of the news at the beginning of the headline. In this case, we are sure that the system headline is much better than the human headline because it gives the exact date of the start of the IPO of Aramco, whereas the human title says that it will start next month, which is a piece of vague information for an investor:

news article:

كشفت مصادر مطلعة لقناة العربيةْ أن الحادي عشر من ديسمبر المقبل هو التاريخ المنتظر لبدء تداول أسهم عملاق النفط أرامكو في السوق السعودية تداول.، ووسط كثرة التكهنات التي دارت حول النسبة النهائية التي ستطرح في الاكتتاب العام الأولي، أفادت مصادر لوكالة رويترز أن الحكومة تتطلع لطرح ٢ % من أرامكو أمام المساهمين من الأفراد والمؤسسات.، كما قالت ثلاثة مصادر مطلعة لوكالة رويترز، إن الحكومة السعودية ستخضع لقيد مدته سنة ...

finarat5:

رويترز: ١١ ديسمبر موعد بدء تداول أسهم أرامكو

Translation: Reuters: December 11 is the date for the start of trading in Aramco shares.

gold:

رويترز: انطلاق اكتتاب الأفراد في أرامكو هذا الشهر

Translation: Reuters: Retail IPO in Aramco begins this month.

Another example shows the power of our model. In this case, the financial Arabic model is able to detect the family name of the energy minister of UAE and uses it instead of just saying the energy minister. This is the highest level we could reach for automatic abstraction. This was possible through pretraining on a custom corpus of Arabic datasets, including several names of CEOs and CFOs from the different financial reports we used in our data training. Therefore the model is used for Arabic names and surnames, and it is also used to match or substitute the name and the function of the person.

news article:

قال وزير الطاقة الإماراتي سهيل المزروعي اليوم السبت إن متوسط سعر النفط بلغ ٧٠ دولارا للبرميل في عام ٢٠١٨.، كانت أوبك اتفقت في ديسمبر كانون الأول مع منتجين كبار

آخرين بقيادة روسيا على خفض إنتاج النفط ٢٫١ مليون برميل يوميا لإحداث توازن في سوق النفط اعتبارا من يناير كانون الثاني.، وقال المزروعي في مؤتمر صحفي بأبوظبي آنحن أمام متوسط حوالي ٧٠ دولارا في عام لخام برنتْ مضيفا أن هذا المستوى سيساعد على ...

finarat5:

المزروعي: متوسط سعر النفط ٧٠ دولارا في ٢٠١٨

Translation: Al Mazrouei: The average price of oil is $70 in 2018.

gold:

أسعار النفط تهبط ٢ % بعد ٩ جلسات تداول صاعدة

Translation: Oil prices fall 2% after 9 bullish trading sessions.

Another example is where the custom Arabic model outperforms the gold by giving more accurate details (verified) from the main article. The gold is so vague and does not explain the announcement exactly, while the finarat5 gives exactly the topic with a statistic supporting this.

finarat5:

رئيس القصيبي للعربية: ٩٤ % من دائني القصيبي يوافقون على إعادة هيكلة الديون

Translation: Al-Gosaibi Chairman for Al-Arabiya: 94% of Al-Gosaibi's creditors agree to restructuring the debts.

gold:

مجموعة القصيبي تكشف للعربية: خطوة واحدة أمام إنهاء المهمة المستحيلة

Translation: Al Gosaibi Group reveals to Al Arabiya: One step towards ending the impossible mission.

7.6 Different Ethics Statements

Data: The pre-training corpus is collected from public domains. However, this is a set of annual financial reports. So, all the copyrights are held by the companies. In addition, the news wires are owned by a large media institution in the Middle East. Therefore, we are not able to make the dataset public.

Energy efficiency: Our models consume significant computational power for pre-training since it takes several days to train a language model. This motivates the technical decision not to start the training from scratch and base our model on another Arabic model.

Tasks: The models have been fine-tuned for financial news text summarisation. We cannot guarantee similar results on general-purpose Arabic content.

Risk The summarisation models generate highly abstractive summaries. In some cases, they may hallucinate producing incorrect or misleading output.

7.7 CO2 Emission Related to Experiments

Experiments were conducted using the Google Cloud Platform in region us-central1, which has a carbon efficiency of 0.57 kgCO_2eq/kWh. A cumulative 1080 hours of computation was performed on the hardware of type TPUv3 Chip (TDP of 283W).

Total emissions are estimated to be 174.21 kgCO_2eq, of which 100 per cent were directly offset by the cloud provider.

Estimations were conducted using the MachineLearning Impact calculator presented in Lacoste et al. (2019).

174.21 kg of CO2eq. is equivalent to: 704 Km driven by an average ICE car, 87.3 Kgs of coal burned [2] 2.9 Tree seedlings sequesting carbon for ten years [3]

Kg CO2 eq. Power consumption x Time x Carbon Produced Based on the Local Power Grid: 283W x 1080h = 305.64 kWh x 0.57 kg eq. CO2/kWh = 174.21 kg eq. CO2

7.8 Summary of the Chapter

In this chapter, we targeted the task of abstractive summarisation on an Arabic news summarisation use case. We continued the training of an Arabic text-to-text model on a corpus of Arabic financial texts that we collected and cleaned ourselves. Our results showed that monolingual financial monolingual models could generate coherent and accurate texts in the Arabic financial domain and could be a good benchmark for financial Arabic NLP. We ended by performing a human evaluation experiment using Best-Worst Scaling (Louviere et al., 2015) and also calculated the Osman readability metric of the generated summaries. This chapter can open the doors to new use cases for custom field text-to-text models. News agencies may use text-to-text models to generate the titles automatically to accelerate the news production pipeline. In addition, text-to-text models can be used to generate highlights (case of multi-document summarisation). Pretrained models could also generate headlines for the most-read articles in the last timeframe and use them as highlights in news-crawling websites and mobile apps.

Acknowledgement: We gratefully acknowledge Lancaster University's support for giving us access to the high-end computing GPU cluster. We also thank the Google TensorFlow Research Cloud TFRC[29] program for the free access to Cloud TPUs V3.8, which was crucial for the pretraining process. In addition, we thank the Google Cloud research team for the 1000 USD GCP credits[30] to perform this research. We also acknowledge the arat5 team for their technical help.

[29] https://sites.research.google/trc/about/

[30] https://cloud.google.com/edu/researchers

Chapter 8

Medium Size Financial Document Summarisation Monitoring: A Case Study on French Companies' Annual Financial Report Summarisation

Following chapter 6, where we addressed the problem of purely extractive long financial document summarisation and chapter 7 where we studied the abstractive summarisation of short financial newswires, in this chapter, we will address the summarisation of medium-sized financial reports. This chapter presents an innovative monitoring study of the evaluation metrics during the task of French Financial Narrative summarisation. In fact, it is important to monitor evaluation metrics during the training process and use them to guide the development of the model. For example, if the model consistently performs poorly on a particular metric, it may be necessary to adjust the model architecture or training process to improve its performance. The main aim of this chapter is to perform a tracking study of the summarisation training process. Our research chapter makes the following contributions:

- We present a literature review about NLP training monitoring and we present the different evaluation metrics we used in this chapter.
- We perform a tracking study within French narrative summarisation. This study monitors evaluation metrics' evolution during the French Financial Narrative Summarisation task. It benchmarks four summarisation techniques (monolingual seq2seq models, multilingual seq2seq models, Encoder2encoder models and encoder2decoder models).
- We describe a set of unsupervised and heuristic rule-based summarisation algorithms applied to financial summarisation.

- We compare supervised and unsupervised models on a French summarisation dataset using Rouge metric variants.
- We perform a statistical significance study to verify the statistical significance of the results.

8.1 Introduction

In this chapter, we present a new approach to monitoring the performance of French financial narrative summarisation systems, in which we focus on monitoring evaluation metrics. We use a dataset of French financial narrative articles presented in section 5.4 and evaluate the performance of different summarisation models using a range of metrics. We consider a range of commonly used evaluation metrics and assess their effectiveness in predicting the quality of the summaries. Our results show that the choice of the model architecture can significantly impact the performance of summarisation models and highlight the importance of carefully selecting and monitoring metrics and reward functions when evaluating summarisation systems. Additionally, we discuss the challenges and considerations involved in assessing summarisation models for financial texts and suggest directions for future research in the NLP field. Our results provide valuable insights for researchers working on French financial narrative summarisation and highlight the importance of carefully selecting and monitoring evaluation metrics. In the second part of this chapter, we will compare a set of transformer-based summarisers with a wide range of unsupervised systems and will end our chapter by performing a set of pairwise statistical tests to evaluate the statistical significance of our results on the different system summaries.

8.2 Background

8.2.1 NLP Monitoring

There is a long-lasting interest in understanding the internal behaviour of black-box neural network and deep learning systems (Alishahi et al., 2017; Pomerleau, 1990; Raghu et al., 2017). Previous editions of the BlackboxNLP Workshop[1] (Linzen et al., 2019; Alishahi et al., 2019; Alishahi et al., 2020; Bastings et al., 2021; Bastings et al., 2022) focus on analysing and interpreting neural networks for NLP. The workshop series published a variety of papers on understanding and elucidating black-box NLP models. For example, Lippincott (2018) presented Vivisect, a toolkit designed to provide a comprehensive and detailed monitoring service across major deep neural network (DNN) frameworks with minimal disruption to research processes. Schröder et al. (2022) investigated using monitoring as a possible solution to mitigate six categories of challenges for verification and validation of machine learning applications

[1] `https://blackboxnlp.github.io/`

during production. To the best of our knowledge, there is no public summarisation system training monitoring study.

8.2.2 Evaluation Metrics Used in this Study

In this chapter, we are using three categories of metrics: n-gram-based metrics such as ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) METEOR (Banerjee et al., 2005), model-based metrics such as BERTScore (Zhang et al., 2020b), Frugal score (Eddine et al., 2021) and Bleurt (Sellam et al., 2020), statistical metrics such as Depthscore (Staerman et al., 2021), BaryScore (Colombo et al., 2021b), and infoLM (Colombo et al., 2021a). All the metrics were previously presented and detailed in section 2.7.

For ROUGE (Lin, 2004), we used four variations: ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-LSum. We used a French stemmer, and we reported the mean of the F1 score of the different variants. The F1 score metric measures the harmonic mean of precision and recall for the generated summary. It is helpful in evaluating the overall quality of the summary and identifying false positives and false negatives. Regarding BERTScore (Zhang et al., 2020b), we used the multilingual large Bart model of Facebook[2] and we used also the multilingual Cased Bert checkpoint. We also skip special tokens while calculating the score. We report the mean of F1 scores. We used the original implementation of the metric.[3]. For CIDEr (Vedantam et al., 2015), we used 4 grams for the cider score calculation. The evaluation is calculated per batch. For Bleurt (Sellam et al., 2020), our implementation clones the original version[4].

For Frugal score (Eddine et al., 2021), we followed the implementation proposed by the authors of the paper on github[5] which is also supported by Huggingface API. We measure two versions of the frugal score in our study. The first one is called frugal score (mover-score) based on this checkpoint[6] which means we are implementing a tiny version of moverscore using bert base model. The second one is called frugal score (bert-score) based on this checkpoint[7] which means we are implementing a tiny version of bertscore using Roberta base model.

To implement BaryScore (Colombo et al., 2021b) in our pipeline, we referred to the original code released by the main authors, which requires a large amount of GPU memory. The code is heavily based on the Optimal Transport for signal, image processing and machine learning python library[8] (Flamary et al., 2021). We use the French version of bert: ‘camembert base’ in our Baryscore implementation. The Baryscore uses the 2D free support Wasserstein barycenters of distributions

[2]https://huggingface.co/facebook/mbart-large-cc25
[3]https://github.com/Tiiiger/bert_score
[4]https://github.com/google-research/bleurt
[5]https://github.com/moussaKam/FrugalScore
[6]https://huggingface.co/moussaKam/frugalscore_medium_bert-base_mover-score
[7]https://huggingface.co/moussaKam/frugalscore_medium_roberta_bert-score
[8]https://pypi.org/project/POT/

introduced by the POT library [9]. For **Depthscore**, the pseudo metric used was introduced in Staerman et al. (2021). **InfoLM** (Colombo et al., 2021a) can be implemented using different measures to calculate similarity. In this study, we used kl[10] , beta[11] , l1 Norm[12], and fisher_rao[13]. The metric will get a discrete reference distribution over the vocabulary and compare it with the hypothesis distribution over the vocabulary. **l1 norm** returns l1 norm between the reference and hypothesis distribution. **kl** returns Kullback–Leibler divergence between the reference and hypothesis distribution. **beta** returns beta divergence between the reference and hypothesis distribution. **fisher rao** returns fisher rao distance between the reference and hypothesis distribution.

8.3 Training Monitoring

In this section, we will cover experiment tracking. We will learn to track experiments by logging and monitoring using TensorBoard or Weights & Biases (W&B). These tools enable us to efficiently host and track experimental results such as loss, learning rate, or other metrics, which helps us optimise model training. We can utilise either TensorBoard or W&B to monitor and optimise our models and display an experiment's outcomes.

The main reasons to monitor an NLP system training are to:

- Understand the accuracy of the predictions: it helps us to figure out what are the different things that we can change that, in turn, affect the final accuracy.
- Prevent prediction errors so that we will know which version of the data and the model gives us the best out of the sample predictions.
- Tweak the model to perform better.
- Learn to estimate the ideal number of epochs to get the best results.

Tracking model training live with W&B: W&B, unlike TensorBoard, provides a dashboard in a cloud platform, and we can trace and back up all experiments in a single hub. The training code is run on our local machine or an HPC cluster, while the logs are kept in the W&B cloud. Most importantly, we can follow the training process live and share the results publicly.

Features of Wandb: Wandb helps to build end-to-end MLOps pipelines with several features such as experiment tracking, dataset versioning via artifact, hyperparameter tuning and model life cycle management. W&B offers a range

[9] https://pythonot.github.io/auto_examples/barycenters/plot_free_support_barycenter.html
[10] https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
[11] https://en.wikipedia.org/wiki/Beta_distribution
[12] https://mathworld.wolfram.com/L1-Norm.html
[13] https://en.wikipedia.org/wiki/Fisher_information_metric

of helpful features, such as automation of hyperparameter optimisation through W&B Sweeps tool. Additionally, it provides system logs related to GPU and CPU utilisation (energy monitoring). The visualisation gives us a summarised performance result for a single run. However the W&B platform allows us to explore the results dynamically by combining a lot of runs on the same graph. Weights & Biases is ideal for experimentation, exploration, and the ability to replicate models in the future. It can also store model checkpoints so that we can recreate the project if needed.

8.4 Transformer Based Summarisation Architecture

Qiu et al. (2018) investigated the effect of incorporating pretrained language models into RNN and proposed a new type of pretrained language model based on recurrent neural networks (RNNs). Their research showed that pre-training an RNN-based model on unlabelled data and then fine-tuning it for a specific downstream task was more effective than directly training a randomly initialised model. In addition, this will reduce the size of labelled training data needed for the finetuning process. Afterwards, several companies such as OpenAI and Google released pretrained language models based on transformers, such as GPT by Radford et al. (2018) and BERT by Devlin et al. (2019). Transformer models proved to be more efficient than recurrent neural networks. This section will present pretrained language models and the two different architectures used for summarisation in this monitoring study.

8.4.1 Pretrained Seq2Seq Language Models

The most effective summarisation techniques utilise the transformer encoder-decoder (TED) architecture, as Vaswani et al. (2017) introduced. As examples of these approaches, we can mention those proposed by Lewis et al. (2019), Raffel et al. (2020), Zhang et al. (2020a) and Zaheer et al. (2021). These models require high computational resources to be trained. In addition, these models have been found to improve SOTA results on sequence-to-sequence modelling tasks. Sequence-to-sequence tasks can be defined as a mapping from an input sequence $\mathbf{X}_{1:n}$ ($\mathbf{X}_{1:n} = \mathbf{x}_1, \ldots, \mathbf{x}_n$) to an output sequence $\mathbf{Y}_{1:m}$ ($\mathbf{Y}_{1:m} = \mathbf{y}_1, \ldots, \mathbf{y}_m$). Thus, a sequence-to-sequence model should be able to calculate the conditional probability distribution of the output sequence $\mathbf{Y}_{1:m}$ given the input sequence $\mathbf{X}_{1:n}$.

$$p_{\theta_{\text{model}}}(\mathbf{Y}_{1:m}|\mathbf{X}_{1:n}).$$

Seq2Seq models could be either monolingual or multilingual. In this chapter, we will use both to see how they evolve during the training process.

8.4.2 Manually Created Encoder-Decoder Models

Rothe et al. (2020) presented in their paper a new way to bypass the time-consuming pre-training process for encoder-decoder models. Rothe et al. (2020) suggested using pretrained encoder and/or decoder-only model checkpoints (e.g. BERT, Roberta, GPT2, GPT3) to construct an encoder-decoder model manually. This paper showed that we could attain similar performance to large pretrained encoder-decoders on various sequence generation tasks while requiring less training effort.

Encoder-only models such as BERT must have an output length predetermined and are not suited for sequence-to-sequence tasks. Decoder-only models can be useful for seq2seq tasks but have certain restrictions in their architecture. To bypass this constraint, we should use an encoder-only model as an encoder and a decoder. Therefore, we manually create a model following the Transformer Encoder Decoder architecture proposed by Vaswani et al. (2017), and then we can train it on custom data. The same idea could be applied to decoder-only models (eg. GPT-2) by adding an encoder only model in order to create a TED model combining a stand-alone encoder (such as BERT) and a stand-alone decoder (like GPT3). The question that arises is how one can use the stand-alone model checkpoints to generate encoder-decoder models and which combinations of these models are most effective for summarisation tasks. In this chapter, we will develop two types of encoder-decoder French summarisation models: the first will use two encoder-only French transformers, and the second will use an encoder-only and a decoder-only French transformer. The used manually created encoder-decoder models are:

- **C2C base** (Camembert2Camembert base): This summariser transformer uses French CamemBERT Base as an encoder and then uses it as a decoder.
- **C2C large** (Camembert2Camembert large): This summariser transformer uses French CamemBERT Large as an encoder, and then it uses it as a decoder also.
- **camembert-gpt2-fr-small**: This summariser transformer uses French CamemBERT Base as an encoder, and then it uses GPT-fr small as a decoder.
- **camembert-gpt2-fr-base**: This summariser transformer uses French CamemBERT Base as an encoder, and then it uses GPT-fr base as a decoder.
- **camembert-belgpt2**: This summariser transformer uses French CamemBERT Base as an encoder, and then it uses BelGPT-2 as a decoder.

8.5 Experiments

8.5.1 Experimental Setup

We ran experiments for each model separately on the French narrative summarisation dataset utilising a single NVIDIA RTX 8000 85Go GPU server. For BARThez, mBART, mBARThez and mT5, we used a learning rate of 5e-5 with train_batch_size and eval_batch_size of 3 and 2, respectively, along with a seed of 42. The Adam optimiser was used in combination with a linear learning scheduler. Specifically, BARThez (base architecture comprising six encoder layers and six decoder layers) was fine-tuned for ten epochs. The best model was selected based on the lowest loss value obtained on the dev set. The duration of each training session was around between 12 and 16 hours. This is justified by the very long and complex process to calculate all the metrics after each epoch especially the statistical-based metrics such as Baryscore and infoLM which requires a lot of time and memory to be calculated and passed to the wandb callback. Technically we have created a network connection between the Lancaster High End Computing (HEC) service[14] and wandb cloud server[15]. Therefore after every epoch we calculate a set of evaluation metrics on the validation sample then we send the result to the cloud server. We have the choice between reporting to wandb every epoch or every step (One epoch is composed of several steps). Afterwards, on the wandb platform we can superimpose several monitoring experiments on the same graph to compare them against each other.

8.5.2 Hyperparameter Choice

The wandb platform has a very useful service called wandb sweeps enabling automating the hyperparameter search. Sweeps can be used to calculate parameter importance and efficiently sample the space of hyperparameter combinations. For hyperparameter tuning we can use three techniques: **Grid search**, **Random Search** and **Bayesian Search**. There is a fourth method named Heuristic Search which relies on experimentation to reach an optimal outcome. It combines exploration and exploitation. In our case, hyperparameter search is a critical part of this experiment. The aim in itself is not to maximise the accuracy but to find the optimal parameters combination that allows executing the training while taking into account the limited GPU memory available (around 80 GB of GPU memory) and the limited access to the wandb platform.

Following our initial experiments using wandb sweeps, we reach a conclusion where the **per_device_train_batch_size** and **num_train_epochs** are the most parameters impacting our memory use. Since the number of epochs is predefined to 10 in our case, we focused our optimisation procedure on choosing the most optimal train batch size. This is the most important hyperparameter since it directly impacts our memory usage. With a batch size of 2, the training requires a higher number of steps and, therefore, longer time, leading to getting timed out from the wandb synchronisation. Using a batch size of 4, the training process becomes very GPU memory-hungry (because we have a large training textual input), leading to getting out of memory for the PyTorch framework we are using. We could not allocate enough memory from the GPU clusters of the HEC. So, the most optimal value that

[14]https://www.lancaster.ac.uk/iss/info/IThandouts/hec/HEC-flyer.pdf
[15]https://www.wandb.ai

worked for our experiments was a batch size of 3. (Detailed hyperparameters are presented in appendix D.1)

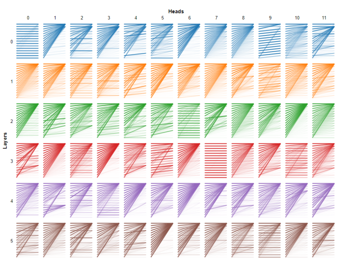
8.5.3 Data Preparation

The dataset used is CoFiF Plus and was presented in Section 5.4 (Zmandar et al., 2022). It is the first financial narrative summarisation dataset providing a comprehensive set of financial text written in French. As a reminder for the reader, the dataset was extracted from French financial reports published in PDF format. It is made up of 1,703 reports from the most capitalised companies in France (Euronext Paris), covering a time frame from 1995 to 2021.

8.5.4 Barthez Model Visualisation

Figure 8.1a, Figure 8.1b and Figure 8.1c show an internal visualisation of attention across all of the model's layers and heads for a particular input ("CAC 40: une hausse prudente avant l'inflation") and output ("CAC40: se rapproche des sommets, gain hebdomadaire de 4,4% ") of the fine-tuned Barthez model on our French financial summarisation dataset. The three figures represent respectively, the encoder, decoder and a cross representation(Encoder-Decoder). The attention heads are organised in the format of tables, where columns show heads and rows indicate layers. The model view (illustrated in Figure 8.1) enables us to browse the attention heads across all layers quickly and to see how attention patterns evolve throughout the model. This view is particular to a specific input and output mentioned above.

(a) Barthez: The model view of the fine-tuned Barthez language model encoder



(b) Barthez: The model view of the fine-tuned Barthez language model decoder

(c) Barthez: A cross model view of the fine-tuned Barthez language model

Figure 8.1: Barthez: The model view of the encoder and decoder of the finetuned language model

8.6 Plots and Analysis of Monitoring Study

8.6.1 French Pretrained Seq2Seq Models

Introduction: In this section, we analyse the evolution of different scores on the validation split during the French narrative summarisation task using two monolingual French models suitable for seq2seq tasks. We focus on evaluating previously described metrics, which are a mixture of commonly used in the field of NLP for summarisation tasks and newly introduced metrics. We aim to understand how the model performs over time and identify data trends or patterns since these models should perform the best on a French summarisation task. The difference between these two systems is that Barthez has a Base architecture and was trained from scratch on a novel French dataset, while mBarthez is trained on the same corpus, but the start point was the last checkpoint of the previously released *mBART* model from Facebook AI. The monitoring plots are presented in figure 8.2.

Figures Description: The figures presented in this subsection show the evolution of the previously described metrics on validation split during the summarisation task. The two systems are trained for around ten epochs. Every epoch is composed of several steps. We calculate the evaluation metric after every epoch, not every step. This is for optimisation purposes. The x-axis represents the number of training steps or iterations, while the y-axis represents the metric value. (Eg. eval/baryscore means that we are reporting the baryscore on the validation split: We are reporting ten measures).

Figure Analysis:

- Global overview: We can easily see an overperformance of Barthez over mBarthez on all the metrics reported in this plot (ngrams, embedding based and statistical-based metrics). This matches the idea of the superiority of monolingual models over multi-lingual models (even if they were more trained on a specific language)

- Interpretation of the Figures: We observe that the Rouge scores (R1, R2, RL, RLsum), Bleurt, meteor, BE, and BA scores increase rapidly in the early epochs, indicating that the model is learning quickly. However, after a certain point, the rate of improvement slows down, suggesting that the model is reaching its performance limit. However, we also observe that the rate of improvement slows down as the model approaches its maximum performance, which is a normal behaviour. Furthermore, we notice that the difference in performance between the two models varies across different metric monitoring plots. The two variants of the Frugal score do not show a significant increase in value, but in percentage, we can see an increase of over 50% since the start of the training. Our question is about the reason for the low start value during the training process. This could be justified by the fact that the model weights proposed

Figure 8.2: Monitoring of French pretrained Seq2Seq model(validation split).

Figure 8.2: Monitoring of French pretrained Seq2Seq model(validation split).

by the author are English-pretrained monolingual models. Nevertheless, the shape of the curve is still promising. For the Baryscore (statistical metric), we see a specific behaviour where the score decreases drastically and stabilises after 1500 steps. In addition, the four variants of infolm show comparable behaviour where there is no trend, and we have random movements (sinusoidal movement: up and down). In this situation, we can not use early stopping, and our final result will depend on when we stop the training. So, we should exclude these metrics from our summarisation use case since they do not follow a clear trend of learning, as they may be non-useful as a reward function for a summariser system. Furthermore, Depthscore (statistical metric) also shows a random behaviour without a clear trend.

- Loss function: the model should minimise the loss to make the prediction close to the true labels. In this case, the loss function for the two models decreases significantly during the first training phase which indicates an improvement in their performance. However, the mBarthez shows a sudden increase after 1.5k steps which means that the model disconverge in the final steps. This may explain the lower performance of this model compared to Barthez.

- Explanation of the Results: The rapid improvement in scores in the early epochs is likely due to the model's architecture (Pretrained black box encoder-decoder), enabling it to learn quickly on generative tasks. However, as the model becomes more sophisticated, it becomes more challenging to continue improving performance. It means it is converging after N epochs. Additionally, the pre-training corpus may justify the differences in performance across the two models. Although Barthez has a BASE architecture and mBarthez has a large architecture, it is clear that Barthez overperforms mBarthez. This is justified by the fact that mBarthez is a continuation of training rather than starting from scratch. It was initially trained on the multilingual mc4 dataset. This multilingual corpus may affect the ability to transfer learning to downstream tasks.

- Implications and future work: our results have implications for designing and evaluating machine summarisation models in NLP. Early stopping may be beneficial to avoid overfitting and wasting computational resources, especially with large models such as mBarthez.

Conclusion: Our analysis of different score evolution during machine summarisation using a French monolingual transformer model provides valuable insights into the performance of NLP models over time. The results suggest that models can learn quickly and perform well on embeddings and n-gram-based metrics. However, careful model selection and training are essential for maximising performance. These findings can inform the design and evaluation of summarisation systems in NLP and provide direction for future research in NLP reverse engineering. This study emphasises also on the importance of using early stopping, and we should opt for implementing the saving of the best weight on the validation set rather than saving the last checkpoint.

8.6.2 Encoder (Camembert) - Decoder (GPT-2)

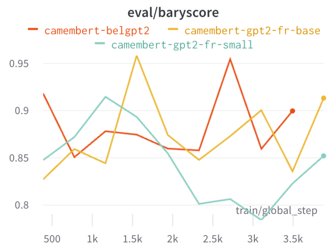
In this subsection, we present the report of monitoring the training of three created transformer summarisers. These models comprise CamemBERT Base as an encoder and three French GPT as used decoders (gpt2-fr-base, gpt2-fr-small and belgpt2). We analyse the evolution of evaluation metrics on the validation split during a French summarisation task for three summarisers we created. The goal is to understand how the model performs over time and identify any data trends or patterns the financial text may relate to. We focus on evaluating previously described metrics, which are a mixture of commonly used in the field of NLP for summarisation tasks and newly introduced metrics. The difference between the three systems is the decoder (GPT2-fr-small, GPT2-fr-base and BelGPT) used to build the transformer summariser. They are not native encoder-decoder models. The monitoring plots are shown in figure 8.3.

Figure Description: The figures presented in this subsection show the evolution of the previously described metrics on validation split during the summarisation task. The three systems are trained for ten epochs. Every epoch is composed of several steps. We calculate the evaluation metric after every epoch, not every step. This is for optimisation purposes. The x-axis represents the number of training steps or iterations, while the y-axis represents the metric value. Each figure corresponds to a different model or experimental setup.

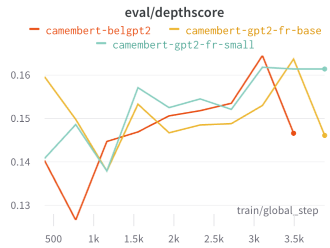
Figures Analysis:

- Global overview: in this case, we can not distinguish the superiority of a model over the two other models. Although camember-belgpt2 showed a slight overperformance, it was not very significant. This may be justified by the fact that the three used summarises are not native summarisation models.

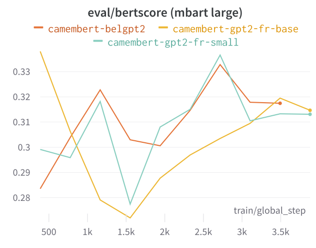
- Interpretation of the figures: overall, we observe that the four variants of Rouge have a plateau curve (there is a small upward trend, but it is not clear due to some fluctuations). It means this technique of summarisation using a non-native encoder-decoder does not favour the improvement of the rouge score, although it may generate text. In addition, we witnessed a steady increase in the Bleurt score, the two variants of Frugalscore and Bertscore, for all models over time. However, the shape of the curve is not parabolic and includes a lot of fluctuations. This suggests that the models improve their summarisation performance as they are trained. However, it is not a significant increase (compared to monolingual models) justifying this task's high computational power requirements. In addition, some fluctuations in the evolution of these metric score curves are worth examining. It means that the improvements are not consistent and continuous. Furthermore, for the four variants of infolm, we see a mean reversion pattern, which means the three systems converge to a



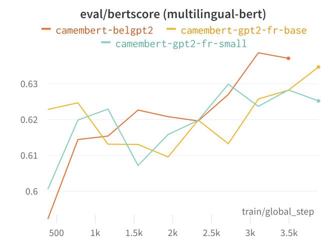
(a)



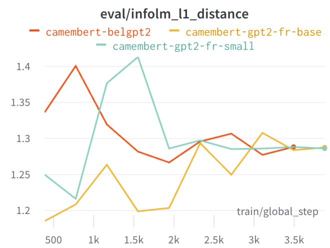
(b)



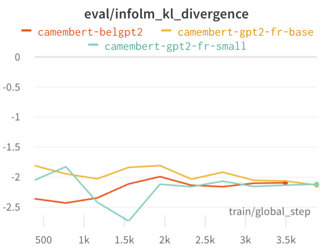
(c)



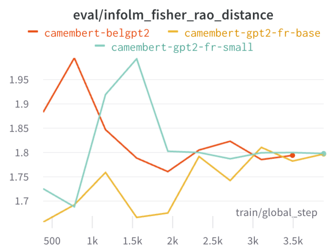
(d)



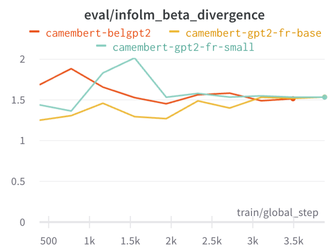
(e)



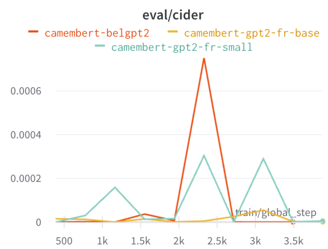
(f)



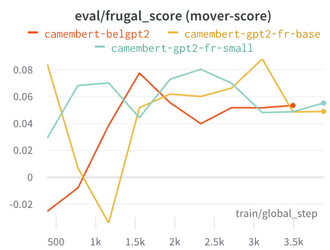
(g)



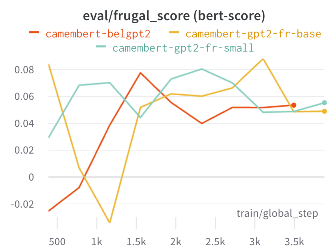
(h)



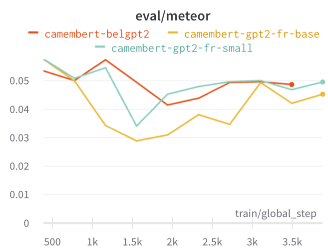
(i)



(j)



(k)



(l)

Figure 8.3: Monitoring of French Encoder (Camembert) - Decoder (GPT) model (validation split).

Figure 8.3: Monitoring of French Encoder (Camembert) - Decoder (GPT) model (validation split).

mean and the standard deviation decreases over time. We can have the same conclusion as the previous section for the fact that infolm does not show a clear trend (some models increase and others decrease). All the models converge to a similar value. So, in the end, we can not distinguish which is the best model using infolm. Nevertheless, the fact that this metric stabilises means we can opt for early stopping if we aim only to optimise this metric. In addition, Baryscore and Depthscore showed a small trend that is not easily identifiable, and there is a high standard deviation towards reaching the final result. Finally, meteor, an ngram-based metric, showed a decreasing curve, which is the opposite to what it is supposed to do during the training.

- Loss function: The loss function on the training data shows a consistent decrease with a smooth learning curve for camembert-gpt2-fr-small and camembert-belgpt2, which converge after 2.5 k steps. For the camembert-gpt2-fr-base, the loss function decreases continuously before starting to increase again after 1,5k steps, which shows again how easy it is to overfit the training set and how important it is to use an optimal number of epochs and apply early stopping to avoid such behaviour.

- Explanation of the results: the previously presented results may be justified by the fact that we are not using a native encoder-decoder pretrained model. Instead, we construct it ourselves using previously pretrained encoder-only and decoder-only models. So, we can not expect as much learning ability as the monolingual seq2seq model. Instead, we can have a decent summariser with less computational power requirement. Therefore, we need to find a good tradeoff between a pretrained seq2seq model (high cost of pre-training) and a manually constructed encoder-decoder model.

- Implications and Future Work: Our results could be used as a proof of concept to construct a French summariser based on two previously released models. The positive side is that the model is learning and can generate coherent text. The negative side is the slow evolution of the learning curve. Future work could explore the impact of changing the encoder from CamemBert to Flaubert, another French encoder-only model.

Conclusion: our analysis shows the start of exploring this summarisation architecture, which will be enhanced by the emergence of new generative models that could be used as a decoder. In this study, we used a GPT-2 version for French. Following the introduction of the GPT-3 French model, the number of parameters was raised from a few million to a few billion. Therefore, we could further enhance the results of this study. In addition, this study enabled us to detect which are the best metrics to use as a reward function if we would like to create our own French Encoder-Decoder summarisers. However, we should mention that this artificial encoder-decoder models are weak learners, although they may be useful if we do not have a monolingual pretrained encoder-decoder model or text-to-text model.

8.6.3 Encoder (Camembert) - Decoder (Camembert)

This section summarises the monitoring of the training of two created transformer summarisers. These models are C2C Base and C2C Large. C2C Base is composed of CamemBERT Base as an encoder/decoder, and C2C Large is composed of CamemBERT Large as an encoder/decoder.

Introduction: In this section, we analyse the evolution of different scores on the validation split during the French narrative summarisation task using two manually created encoder-decoder French models suitable for sequence2sequence tasks. We focus on evaluating previously described metrics, which are a mixture of commonly used in the field of NLP for summarisation tasks and newly introduced metrics. We aim to understand how the model performs over time and identify data trends or patterns. The difference between the two systems is the size of the CamemBERT model. CamemBERT is a state-of-the-art language model for French based on the RoBERTa architecture pretrained on the French subcorpus of the newly available multilingual corpus OSCAR. The monitoring plots are shown in the figure 8.4.

Global overview: We can see a small overperformance of C2C base of C2C large on all the metrics reported in this plot (nrams, embedding and statistical-based metrics). This is opposed to the idea that large models will perform better than base models. In addition, we have a strange pattern of the plots regarding C2C large favouring the idea that it is not a suitable French summarisation model.

Figure Description: The figures presented in this subsection show the evolution of the previously described metrics on validation split during the summarisation task. The two systems are trained for around ten epochs. Every epoch is composed of several steps. We calculate the evaluation metric after every epoch, not every step. This is for optimisation purposes. The x-axis represents the number of training steps or iterations, while the y-axis represents the metric value. (Eg. eval/baryscore means that we are reporting the baryscore on the validation split, eval/rouge1 means that we are reporting the rouge1 variant on the validation split). We are reporting ten measures. Each figure corresponds to a different model or experimental setup.

Interpretation of the Figures: We observe that the Rouge scores (R1, R2, RL, RLsum) show a clear overperformance of the C2C Base over C2C large with an upward trend although the curve is not smooth. For the embedding-based metrics (Bertscores, Frugal scores and Bleurt), the C2C Base shows a flat curve. In practice, there is a slight increase that is unclear due to scale and due to the fact that we are superimposing two graphs with high standard deviation on the same plot. For infolm, our plots confirm the previous findings that these metrics do not improve or converge to zero by transfer learning (This may be justified by the fact that they are statistical-based metrics rather than ngram or embedding-based ones or they may be non-useful or non-relevant for evaluating text summarisation). For baryscore and depthscore, as stated previously, we do not have a smooth curve that enables us to interpret the evolution of the metric. Finally, for cider and meteor we do not recognise a clear improvement of these metrics during the training process.

Figure 8.4: Monitoring of French Encoder (Camembert) - Decoder (Camembert) model (validation split).

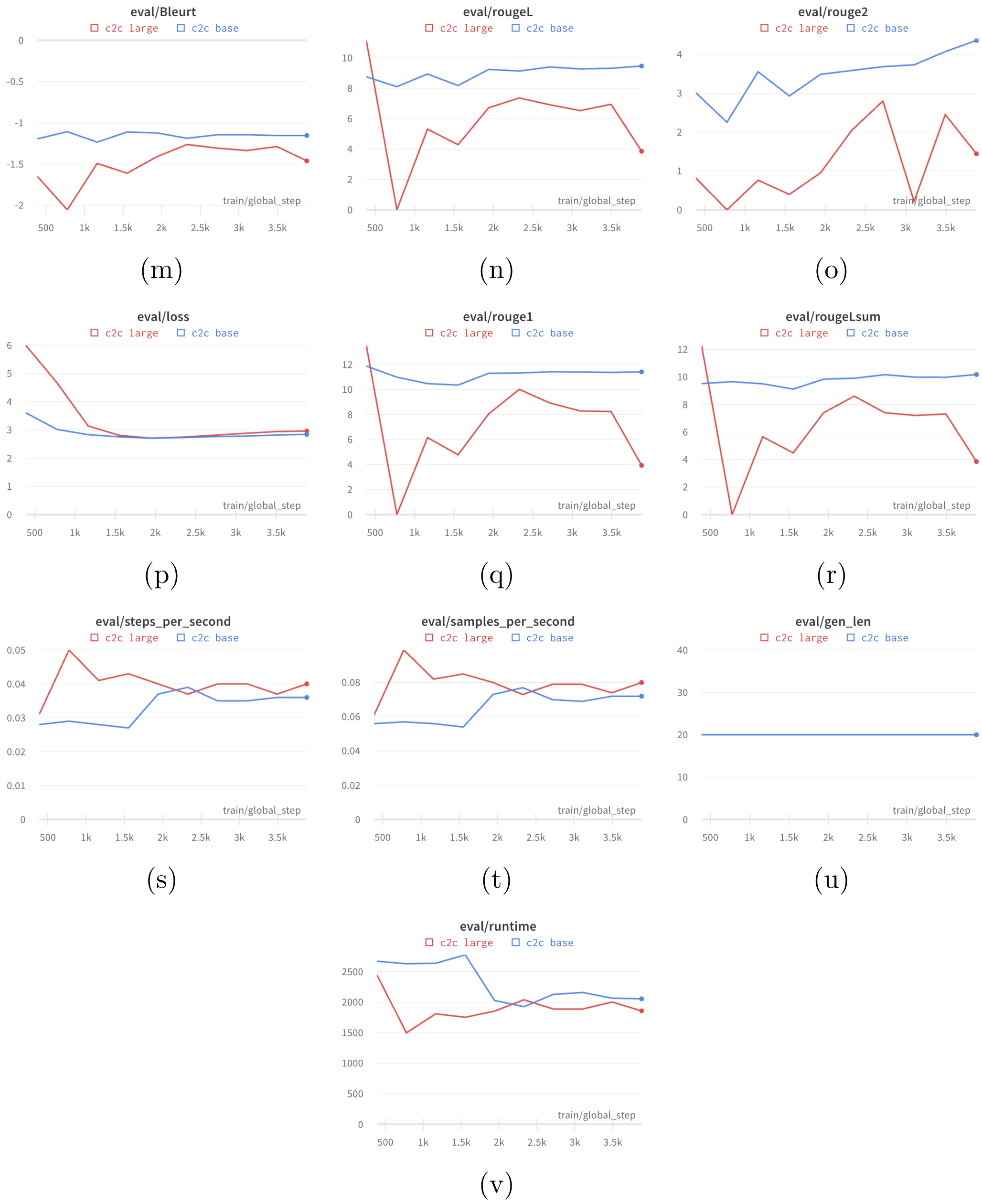


Figure 8.4: Monitoring of French Encoder (Camembert) - Decoder (Camembert) model (validation split).

Loss function: The loss function on the training data shows a flat curve for C2C base (due to scale) and a decreasing trend for C2C large with an overperformance for C2C base. The models are improving through training, but the rate is slow motivating the hypothesis that they are weak learners.

Conclusion: Contrary to the findings of "scaling laws" (Brown et al., 2020) regarding model size and pretrained transformer models, our results did not consistently increase downstream performance when comparing large to base models. Tay et al. (2023) have previously pointed out that good pre-training performances do not always lead to good downstream scores for mT5 models, so we believe that in our case, the lack of improvement may be due mainly to the weakness of this encoder-encoder architecture on a financial text corpus. Although using two French pretrained encoder-only models could be used to train a summariser. However, it is clear that this architecture is non-efficient for French financial summarisation and does not deliver promising results for transfer learning.

8.6.4 Multilingual Pretrained Seq2Seq Summarisers

Introduction: Pretrained multilingual models were trained on vast amounts of text in multiple languages. They are designed to understand and generate text in multiple languages and can be fine-tuned for text summarisation on a French language-based task. The monitoring plots are shown in the figure 8.5.

This section summarises the monitoring of the training of several pretrained multilingual seq2seq models. We monitor mT5, ByT5, mBART and xprophetnet. We analyse the evolution of F1 scores of several categories of metrics on the validation split during the summarisation task using a Weight and Bias academic account. We aim to understand how the multilingual models behave during the training phase.

Figure Description: The figures presented in this subsection show the evolution of the previously described metrics on validation split during the summarisation task. The four systems are trained for around ten epochs. Every epoch is composed of several steps. We calculate the evaluation metric after every epoch, not every step. This is for optimisation purposes. The x-axis represents the number of training steps or iterations, while the y-axis represents the metric value. Each figure corresponds to a different model or experimental setup.

Figure Analysis:

- Interpretation of the Figures: If we exclude the mBART model (high standard deviation), We observe that Rouge scores (R1, R2, RL, RLsum), Bleurt, meteor, cider, Frugal scores, depth score, Bert scores do not increase significantly during the training process epochs, indicating that the model is not learning efficiently. The stagnation in these metrics values is not due to scale issues because we tested without mBART model.

- mBart is the largest model. It has 12 encoder and 12 decoder layers and 610M parameters. It was trained on the common crawl (CC25) corpus. mBART

Figure 8.5: Monitoring of multilingual French Seq2Seq models (validation split).

Figure 8.5: Monitoring of multilingual French Seq2Seq models (validation split).

showed strange behaviour where we had a sudden pick in the negative direction. However, mBart has shown previously a decent capacity for transfer learning.

- Generated length: mBart overperforms all the other multilingual models since it reached 200 generated words on the validation split. This enhances the possibility of generating fully abstractive summaries of 200 words using a black box model.

- Explanation of the Results: These flat curves show how difficult some multilingual models can learn. However, we allocated more than 85 GB of GPU memory and implemented deep work to choose hyperparameters. This is justified because the multilingual models are pretrained on more than 100 languages. Hence, they may be helpful for general-purpose corpus in French. However, for specific contexts, such as publicly listed company financial reports, they did not show promising behaviour.

Conclusion: Our analysis does not show promising results for the ability of multilingual models to perform transfer learning on French tasks, especially in such a technical corpus.

8.6.5 Energy Monitoring Study

In the appendix D.2, we included graphs that monitor the GPU and system usage on the Lancaster HPC GPU cluster during the monolingual SeqSeq training process. The aim is to emphasise on the energy issues related to NLP training systems. The purpose of these graphs is to provide insights into how the system resources are utilised during the training task and to identify potential performance bottlenecks.

Graph 1 GPU power usage (in watts and percentage): These graphs show GPU power usage over time during the NLP training task (the y-axis represents the watt power) and the usage of GPU in percentage (The y-axis represents the percentage of GPU utilisation). In both cases, the x-axis represents the training time in hours. We observe a cyclical/periodic pattern in the graphs. The maximum reached power usage is 200 Watts. The maximum percentage of GPU usage is around 80%. A period T is the required time for one complete cycle to pass a given point. We have a constant period for all the monitoring graphs.

We observe that the GPU usage gradually increases as the training progresses and peaks at around 12 hours. After reaching its peak, GPU usage starts to stabilise and fluctuate slightly. This suggests that the model is utilising the GPU resources efficiently but that there may be room for improvement in optimising the training process to minimise the duration of the training task.

Regarding the GPU temperature, the fluctuation follows the fluctuation of the GPU system's use, peaking at 60 degrees Celsius. The System Usage graph displays the system usage over time during the NLP training task. The y-axis represents the percentage of system utilisation, while the x-axis represents the training time in hours.

Process GPU memory allocated: We observe that the GPU memory allocated to large models is significantly higher than that allocated to Base and Small, which justifies that sometimes we were obliged to train only small and Base models. Reaching large architectures is not very trivial regarding memory availability and hyperparameter tuning. We also have the same behaviour for the network traffic in bytes, where the large models require much more network traffic, and the difference is significant. Therefore, we may need to re-evaluate the assumption that we should go for larger models. There is a trade-off to make. This study shows that larger models consume much more energy but do not always generate better results. This raises again the issue of the cost of training NLP models on generative tasks. Finally, we can see that we have a constant disc utilisation (around 20%), a typical pattern among all the systems we trained.

Implications and Future Work: Our results have implications for designing and optimising NLP training tasks on HPC GPU clusters. Specifically, carefully monitoring resource utilisation patterns can help identify potential bottlenecks and guide optimisation efforts. Future work could explore the use of more advanced monitoring techniques, such as real-time resource allocation and task scheduling, to further optimise NLP training on HPC GPU clusters.

Conclusion: Our analysis of GPU and system usage during an NLP training task on an HPC GPU cluster provides valuable insights into efficiently utilising system resources. The results suggest that careful monitoring of resource utilisation patterns can help identify potential bottlenecks and guide optimisation efforts. These findings can inform the development of more efficient NLP training strategies on HPC GPU clusters and provide direction for future research.

8.7 Unsupervised Financial French Report Summarisation

Unsupervised extractive document summarisation attempts to identify the most significant sentences in a document without needing labelled data. In the preceding section, we examined the behaviour of several metrics' behaviour while training encoder-decoder models. In this section, we will present a few unsupervised techniques used on our French datasets then we will compare them with the pretrained seq2seq models (Barthez, mBarthez, mT5, ByT5, Xprophetnet) that we already trained in the monitoring study.

Unlike the UK reports, French annual reports follow a shared common structure recommended by the financial regulator AMF. It is not a form to fill out like the 8-K or 10-K forms in the US, but we are still able to detect some common patterns. Thus, we need to explore unsupervised summarisation techniques on French annual reports. Knowing the structure can quickly help us leverage rule-based extractive scripts or strong unsupervised baselines. The French ones follow a clear structure. However, they vary somewhat from US reports, where the SEC (Security Exchange

Commission) requires firms to fill out predefined forms with special markup tags. However, detecting a common structure with good analysis is still feasible for French reports. Shortly, the AMF will adopt the XBRL reporting so that these unsupervised methods will get more robust, especially the rule-based extraction methodology.

8.7.1 Rule Based Summaries

A rule-based extractive summariser based on heuristic rules would take text input and generate a text summary by applying a set of heuristic rules. These rules would be designed to identify the essential information in the text and then extract it to create a summary. The advantage of this approach is that it is relatively simple to implement, can be easily customised to different domains or applications, and does not require much computational memory. Additionally, this approach could generate summaries of texts that could be better structured or well-organised, which would be difficult for other methods. The disadvantage of this approach is that the rules to generate the summaries would need to be manually created, which could be time-consuming and error-prone.

In our case, this unsupervised method aims to extract the chairman's highlights, the financial highlights, and the perspectives chapter and put them together in one system-generated summary. In the case of the French annual reports, this is an efficient strategy because of how the corpus was prepared (see section 5.4) and because of the nature of French reports, which have a typical repetitive pattern that helps identify the start of certain parts. Therefore a well-researched algorithm could have a promising extractive accuracy. To avoid a reverse engineering process, we prepared two versions: one with and one without chairman highlights. The one reported in the results is without chairman highlights. This is to avoid the high accuracy and matching that we obtained when putting the three parts in the system summary. In addition, this would avoid using the same algorithm previously used to label the corpus.

8.7.2 Lead 200

The LEAD-k method is a widely used standard for extractive summarisation, which involves selecting the initial k sentences from the original text as the summary (Yang et al., 2020b; Tang et al., 2022a). The k-value, typically 1, 5 or 10, determines the length of the summary in terms of sentences. This is based on the idea that the initial sentences of the text are the most informative and significant, thus making them the most suitable choice for a summary. LEAD-k summarisation is often chosen as a benchmark for testing the performance of other extractive summarisation techniques due to its straightforward implementation and easy assessment against other methods (Perez-Beltrachini et al., 2021). In this section, we are exploring the efficacy of Lead-k for summarising long financial documents in French by implementing it as a baseline for our experiments and we have been impressed by the results. We can confidently say that LEAD-200 (the first 200 tokens in our case) is a robust benchmark for summarising French annual report text, regardless of the document size or type. This is also motivated by the fact that most French companies start their report with a quick highlight that gives a detailed overview summary of the activity during the last tax year. This practice is highly recommended by the French regulator AMF.

8.7.3 First Paragraph

The first paragraph of an annual financial report is often used as a strong benchmark for summarisation because it covers a general overview of the text and its main topics. It is concise and straightforward, making it an ideal candidate for a summary. Most French reports start with a general highlight section. Additionally, the first paragraph represents the text as a whole, indicating its overall tone, style, and content. Main features of the first paragraph: **Relevance**, **Conciseness**, **Representativeness** and **Consistency**. The consistency in annual financial reports makes it easier to understand the main points, but it's important to note that there are better summary benchmarks than this. Financial annual reports are typically structured consistently and predictably, with the first paragraph serving as an introduction to the main content. This consistency makes using the first paragraph as a strong summarisation benchmark. It is also essential to know that using the initial paragraph as a summary reference point is not always the most suitable option because the key performance indicators will be detailed in the further sections (especially the financial highlights section) in a report. Different techniques, like keyword extraction or graph-based summarisation, could be more effective depending on the particular text and the task of summarisation. The selection of the summary benchmark should be contingent on the individual requirements of the summarisation task and the features of the text being summarised. The efficiency of the first paragraph will be evaluated on a case-by-case basis.

8.7.4 Graph Based Summarisation

Following Gokhan et al. (2022), we present an example of Graph-Based unsupervised summarisation applied to our French financial corpus. This method calculates sentence similarities to assign vertex and edge weights. To do so, we use traditional graph ranking algorithms with recent sentence embedding models and sentence features in order to improve how sentence centrality is determined. This system consists of four steps: initially, sentence features are calculated to determine the vertex weight; second, SentenceBERT is used to generate sentence embeddings and measure their similarities; third, a graph is created by contrasting all the pairs of sentence embeddings; lastly, sentences are ranked based on their degree centrality in the graph.

Financial Report $\rightarrow$ $Preprocessing\_and\_cleaning \rightarrow sentence\_embeddings \rightarrow Graph(matrix) \rightarrow Ranking \rightarrow Best\_N\_sentence\_selection$

We first compute the scores for each sentence feature to represent its importance in the document. This step is achieved using the sentence transformers library[16] to calculate the embedding graph of the french annual report. The sentence-transformers model maps sentences and paragraphs to a 768-dimensional dense vector space and can be used for tasks like clustering or semantic search. Once this step is complete, we take advantage of Sentence-BERT[17] created by Reimers et al. (2019) to obtain sentence embeddings to capture the meaning of the sentences more effectively. This allows us to build a graph with edges that depict semantic similarities. Subsequently, an undirected graph is produced, which considers both the significance of each sentence and their mutual similarity. The sentence transform models can be one of the following choices: Monolingual or Multilingual versions. In this study, we propose two versions: one with 'roberta-base-nli-stsb-mean-tokens' and the second with 'distiluse-base-multilingual-cased-v1'. In fact, **distiluse-base-multilingual-cased-v1** is a multilingual knowledge distilled version of the multilingual Universal Sentence Encoder. It Supports 15 languages: Arabic, Chinese, Dutch, English, French, German, Italian, Korean, Polish, Portuguese, Russian, Spanish, Turkish. (Reimers et al., 2020). Lastly, we use a ranking system on the resulting graph to identify the most significant sentences in a document. The algorithm 8 summarises the process described above.

ALGORITHM 8: Main Function for Graph-Based Unsupervised Summarisation

Input: French financial report

Output: Graph-based system summary

Data: Corpus of financial reports from companies listed in CAC40 and CAC20

```
report ← cleanDocument(report)                                  // Clean financial report
sentences ← sent_tokenize(financial report)                     // Sentence Tokenisation
sentenceRankList ← allReportSentenceRanking(sentences, corpus)  // Returns a list of Corpus Sentence Rankings
similarityMatrix ← createGraph(sentences)   // Create Similarity Matrix (Graph) Using Cosine Similarity
Rank ← HighestSimilarityRank(similarityMatrix, sentenceRankList)  // This function creates a new list containing the highest similarity rank
sentenceNumberInSummary ← 5
System_Summary ← createSummary(sentences, Rank, sentenceNumberInSummary)  // Create Summary by Picking Top N Sentences
return System_Summary
```

8.7.5 Bert Clustering Summarisation Technique

Based on (Liu et al., 2019a), we developed a Bert-based clustering unsupervised financial summariser. Algorithm 9 shows in detail the steps to generate a summary using multilingual Bert for clustering. The biggest limitation of this methodology is

[16] https://pypi.org/project/sentence-transformers/

[17] https://www.sbert.net/

the limited number of tokens (512) that Bert could accept as input. While creating the embeddings, Bert model produces 12 layers of latent vector, the 'last layer' is used here. We use the k-means algorithm for clustering.

ALGORITHM 9: Bert-Based summarisation algorithm

Input: Paragraph

Output: Summary

1 Tokenise paragraph into sentences.

2 Format each sentence as Bert input format([add [CLS] and [SEP] tokens), and Use Bert tokenizer to tokenise each sentence into words.

3 Call multilingual Bert pretrained model, conduct word embedding, and obtain embedded word vector for each sentence. (The Bert output is a 12-layer latent vector);

4 Use only the last layer of latent vector

5 Apply a pooling strategy to obtain sentence embedding from word embedding. It could be the mean or max of all word vectors;

6 Obtain sentence vector for each sentence in the paragraph, apply Kmeans, Gaussian Mixture, etc to cluster similar sentences;

7 Return the closest sentence to each centroid (euclidean distance) as the summary, ordered by appearance;

8.7.6 LexRank

LexRank is a graph-based unsupervised method for computing relative importance in extractive summarisation (Ge et al., 2021). This study proposes two Lexrank systems: classical and continuous Lexrank. The first is the classical version of the Lexrank algorithm; the second is a version that extracts continuous N sentences. We generate a five-sentence summary since the average gold standard summary on the test set is composed of five sentences. We used French stopwords. In addition, we set a maximum compression ratio of 8%. On our test split, we have an average 6% compression ratio. This graph-based model is employed for various NLP applications and uses centrality to rank sentences. The process involves randomly navigating through the graph, in which each lexical unit, or vertex V, is represented by a sentence. Here, the similarity score between two sentences x and y is given as:

$$idf-modified-cosine(x,y) = \frac{\sum_{w\in(x,y)} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{x_i\in x}(tf_{x_i,x} idf_{x_i})^2} \times \sqrt{\sum_{y_i\in y}(tf_{y_i,y} idf_{y_i})^2}}$$

where $tf_{w,s}$ is the number of occurrences of word w in sentence s, and idf_w is the inverse document frequency of word w .
Sentence Relevance Score of S is given as:

$$Score(S) = \frac{\sum_{Unigrams\in S} Unigrams}{\sum_{Unigrams\in Document} Unigrams}$$

Manual Evaluation: Lexrank classical generated coherent system summaries. There is an accurate detection for the highlights section ("Faits marquants"), the chairman highlights and the financial highlights. We have a high correlation with the technique used to create the dataset. However, this human evaluation should be confirmed by an automated metric, which will probably penalise these summaries because they are not a continued extracted part, knowing that the corpus is composed of continuously extracted parts of the financial annual report. Regarding the continuous Lexrank, it is a very weak baseline. It did not generate coherent and long enough summaries.

8.7.7 Word Frequency Algorithm

We end our study by preparing another unsupervised method called word frequency algorithm. This method starts by tokenising text into words, we stem words and remove stopwords then we compute word frequency for each word in the paragraph and store it in a table. This will enable us to score each sentence according to the sum of the word frequencies of the words in the sentence: total word value / word count . Then, we have to create a threshold that enables us to determine whether we will include a sentence or not. The threshold could be the average sentence score = total sentence score / sentence number. Finally we had to generate the summary by including the sentence whose score is greater than the fixed threshold. The algorithm 10 explains the steps of this word frequency algorithm.

ALGORITHM 10: Word Frequency Algorithm

Input: Input paragraph
Output: Summary

1 Compute word frequency for each word in the paragraph;
2 Score each sentence according to the sum of the word frequencies of the words in the sentence: total word value / word count;
3 Generate a threshold for sentence selection by computing the average score of all sentences)total sentence score / sentence number);
4 Select sentences with scores above the threshold as the summary;

8.8 RL Summarisation Systems

Li et al. (2018c) proposed an Actor-Critic based Training Framework for abstractive text summarisation. This paper outlines a training system for neural abstractive summarisation that employs actor-critic techniques from reinforcement learning. Regular neural networks only work to maximise the chance of getting accurate summaries, but this often results in poor quality or incorrect sentences. To address this issue, our French RL financial summariser implements an actor-critic tactic to better the training process. We first trained a custom word embedding model on our dataset and a custom vocab model. We used the French word embedding trained in chapter 4. We designed two neural networks (Actor and Critic) to create an extractive RL summariser system. We based our French RL summariser on the approach proposed by Zmandar et al. (2021b). The extractor agent is an lstm neural network with 256 hidden layers. We used a learning rate of 0.001 with a lr decay 0.001, a batch size of 16. The Adam optimiser was used. We trained the actor alone in a bidirectional manner using the previously trained French word embeddings (dimension: 300). The reward function used in the actor-critic loop was rouge 2. All the RL pipeline was trained on an Nvidia GPU, and it took around 5 hours. Semantically, the RL systems generate qualitative coherent French text language that is easy to understand and that summarises in a good and efficient way the annual report. They may not outperform the transformers in terms of automatic evaluation because transformers extract a bloc, and transformers have a higher number of parameters and an advanced architecture. However, RL systems still have a very decent output. The fluency and coherence of RL systems on the French dataset outperformed the result we got previously on the English dataset. This is justified by the high quality of the French dataset, where the gold standards were all manually checked. The qualitative evaluation shows very promising results.

8.9 Baselines

We used three baseline summarisers: POLY (Litvak et al., 2013b), Lexrank and TextRank (Mihalcea et al., 2004) and one Topline summariser —MUSE (Litvak et al., 2013a) which was used previously for the UK financial narrative summarisation dataset. See (El-Haj et al., 2020e) for more details on the baseline summarisers. MUSE is a supervised financial system summariser trained on an English corpus. POLY is another unsupervised summariser model used to summarise financial annual reports. For textrank[18] we used the implementation of Barrios et al. (2016).

8.10 Results

8.10.1 Generated Summaries

In our monitoring study, we evaluated all the supervised models on the validation split. In this section, we will select a few supervised models (monolingual and multilingual pretrained language models) to be compared against unsupervised baselines or benchmarks already presented but on the testing split. We selected the monolingual and multilingual pretrained language models (Barthez, mBarthez, mT5, ByT5, Xprophetnet) that we already trained in the monitoring study, and we tried to investigate if there is a real scientific interest in performing transfer learning on the French reports or if unsupervised systems could do the job, especially since

[18] https://pypi.org/project/summa/

the dataset has common recurrent patterns. Also, this evaluation aims to evaluate quantitatively the unsupervised models. In total, we present 18 system summaries. Barthez French model showed high ability in summarisation by matching exactly the gold standard summaries. However, as it is a Base model based on 'Bart' model architecture, it does not have the ability to generate the whole summary. So, our work was to continue the extraction to make it match at least 200 words, which is the average length of a summary on the corpus.

8.10.2 Manual Interpretation of the Generated Sentences of Models

As opposed to some suppositions, the French models generated a very coherent text which matches precisely the gold standard in several cases. Surprisingly, the 'Barthez' model generated extractive text by generation. This is the best thing that we can opt for. The negative aspect is that such models have limited generation ability (the maximum number of generated tokens is limited). The trained French transformers are generating extractive summaries with high accuracy of the start token of the summary. It means the transformer is detecting the common and repetitive patterns that were used to construct the dataset. In other terms, if we manage to generate longer outputs, we can directly use this output as a system summary without post-processing as we did for the English work. For the moment, fine-tuned French monolingual models can not generate long sequences (more than 200). So, in the case of French narrative summarisation, we have to continue extracting the missing tokens to reach the 200-word summary standard we opted for. In some cases, such as mT5, we already have more than 200 words generated by the model. Table 8.1 shows some statistics about the behaviour of the models on the testing split during the text generation process. We report the Bertscore, the generated length, the loss and the predict_runtime in seconds.

	barthez	mBarthez	mT5	byT5	xprophetnet
Bertscore	0.5217	0.5121	0.4800	0.5491	0.5218
gen_len	123.17	165.61	252.53	254.83	127.36
loss	6.52	7.67	8.18	13.51	12.62
predict_runtime (S)	257.39	524.17	396.95	317.27	578.54

Table 8.1: Statistics about the generated French system summaries

8.10.3 Quantitative Evaluation

Extractive summarisation is often evaluated using *ROUGE* metrics (Kiyoumarsi, 2015). The ROUGE[19] measure finds the common unigram (ROUGE-1), bi-gram

[19] https://github.com/google-research/google-research/tree/master/rouge

(ROUGE-2) and largest common sub-string (LCS) (ROUGE-L) between the ground-truth text and the output generated by the model and calculates respective precision, recall and F1-score for each measure. For the entire dataset, we evaluate standard ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 (Lin, 2004) against all the different gold summaries. To evaluate the generated system summaries against the gold standard summaries, we used the Java Rouge (JRouge2.0)[20] package for ROUGE, using multiple variants (i.e. ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4). (Ganesan, 2018). This package includes a French stemmer and enables the evaluation of summaries in the French language. A good system summary should maximise the average rouge metric with all the gold standards provided. To calculate the rouge score, we used French stop words and a French Tagger. We did not use a French stemmer to make the evaluation rigorous. We report computed F1-score, which is the harmonic mean between precision and recall.
Results: Tables 8.2 and 8.3 present the results of rouge1, rouge2, rouge3 and the standard deviations of all results, respectively. In addition, Tables 8.4 and 8.5 present the results of rouge S1 and rouge S2 and the standard deviations of all results, respectively. Tables 8.6 and 8.7 present the results of rouge-L and rouge-SU4 and the standard deviations of all results, respectively. Table 8.2 shows the performance of different models on the French financial summarisation task based on three unigram variants of rouge: R-1, R-2, and R-3. For each of these, there are three corresponding values: R: Recall, P: Precision, and F1 score. XPROPHETNET and mBARTHEZ are the best-performing models overall, with relatively high scores across all metrics. This may be justified by the fact that these two models are large models and that we are only using rouge scores for evaluation. Then, we have the rule-based algorithm that ranks third. The score of the rule-based algorithm is very biased because this script is a kind of reverse engineering for the dataset creation process. In other terms, if it manages to extract text, we have a very high probability that it matches the gold standards. Otherwise, it does not generate anything and keeps the system summary blank, unlike other models, which will always generate output. If the output does not match the gold standard, it will highly penalise the rouge score. We can confirm this by the high standard deviation of the scores of rule-based reported in the table 8.3. Also, by checking the scores manually, we see that they vary from 0 to more than 0.7 for rouge2, which confirms our hypothesis.

Table 8.4 reports the results of the Rouge S1 and Rouge S2 evaluation metrics for various text summarisation models. The table reveals that XPROPHETNET, MBARTHEZ, RULEBASED and Barthez are the top-performing models regarding Rouge S1 and Rouge S2 scores. These models have achieved the highest scores in both metrics, indicating that they are successful in generating summaries that are similar to the reference gold summaries(between one and three gold standards). On the other hand, LEXRANK-CONTINUOUS is the worst-performing model, with the lowest scores in both Rouge S1 and Rouge S2. This also confirms the efficiency of French monolingual models.

[20] `https://github.com/kavgan/ROUGE-2.0`

model	R-1 / R	R-1 / P	R-1 / F1	R-2 / R	R-2 / P	R-2 / F1	R-3 / R	R-3 / P	R-3 / F1
XPROPHETNET	0.515	0.508	0.472	0.392	0.403	0.366	0.362	0.378	0.341
MBARTHEZ	0.495	0.493	0.457	0.364	0.387	0.347	0.332	0.361	0.320
RULEBASED	0.425	0.525	0.401	0.374	0.382	0.346	0.363	0.353	0.334
BARTHEZ	0.495	0.490	0.452	0.363	0.381	0.340	0.329	0.353	0.312
BYT5	0.473	0.458	0.429	0.334	0.337	0.307	0.301	0.307	0.277
MT5	0.463	0.450	0.421	0.315	0.329	0.296	0.278	0.296	0.264
FR-RL-SUMM	0.480	0.473	0.435	0.318	0.327	0.294	0.273	0.283	0.253
LEAD-200	0.448	0.420	0.399	0.302	0.299	0.273	0.265	0.265	0.240
FIRSTPARAGRAPH	0.367	0.509	0.344	0.301	0.335	0.269	0.283	0.297	0.250
GRAPH-ROBERTA	0.604	0.227	0.289	0.386	0.163	0.201	0.326	0.144	0.175
BERT-CLUSTERING	0.232	0.501	0.287	0.143	0.309	0.171	0.118	0.246	0.137
GRAPH-DISTILIUSE	0.558	0.208	0.266	0.325	0.139	0.170	0.261	0.116	0.140
TEXTRANK	0.364	0.385	0.346	0.163	0.185	0.158	0.112	0.135	0.111
LEXRANK	0.498	0.217	0.258	0.265	0.125	0.146	0.202	0.099	0.114
MUSE	0.376	0.294	0.297	0.183	0.145	0.143	0.130	0.103	0.101
POLY	0.346	0.352	0.317	0.148	0.156	0.134	0.096	0.104	0.086
WORDFREQUENCY	0.732	0.151	0.210	0.469	0.096	0.132	0.361	0.077	0.104
LEXRANK-CONTINUOUS	0.146	0.439	0.164	0.056	0.202	0.066	0.030	0.121	0.036

Table 8.2: Reporting Rouge scores : Rouge R1 R2 and R3

model	R-1 / R	R-1 / P	R-1 / F1	R-2 / R	R-2 / P	R-2 / F1	R-3 / R	R-3 / P	R-3 / F1
XPROPHETNET	0.25	0.28	0.22	0.31	0.32	0.28	0.32	0.34	0.30
MBARTHEZ	0.24	0.27	0.22	0.31	0.32	0.28	0.32	0.34	0.29
RULEBASED	0.39	0.30	0.35	0.39	0.34	0.36	0.40	0.35	0.36
BARTHEZ	0.23	0.27	0.21	0.30	0.320	0.27	0.31	0.34	0.28
BYT5	0.24	0.24	0.20	0.30	0.29	0.26	0.32	0.31	0.27
MT5	0.22	0.25	0.20	0.28	0.29	0.25	0.29	0.31	0.27
FR-RL-SUMM	0.19	0.25	0.17	0.23	0.28	0.22	0.23	0.28	0.22
LEAD-200	0.23	0.23	0.19	0.28	0.26	0.23	0.29	0.28	0.25
FIRSTPARAGRAPH	0.37	0.24	0.31	0.38	0.31	0.33	0.39	0.33	0.34
GRAPH-ROBERTA	0.19	0.17	0.16	0.27	0.15	0.16	0.28	0.15	0.16
BERT-CLUSTERING	0.18	0.23	0.17	0.20	0.27	0.19	0.20	0.29	0.20
GRAPH-DISTILIUSE	0.19	0.16	0.15	0.25	0.14	0.14	0.27	0.13	0.14
TEXTRANK	0.13	0.19	0.12	0.14	0.19	0.13	0.14	0.18	0.14
LEXRANK	0.20	0.16	0.14	0.23	0.13	0.13	0.24	0.12	0.13
MUSE	0.17	0.19	0.14	0.19	0.17	0.15	0.19	0.16	0.14
POLY	0.13	0.16	0.10	0.13	0.13	0.09	0.12	0.12	0.09
WORDFREQUENCY	0.15	0.16	0.16	0.18	0.12	0.11	0.18	0.10	0.10
LEXRANK-CONTINUOUS	0.13	0.23	0.11	0.07	0.22	0.07	0.06	0.21	0.06

Table 8.3: Reporting the Standard deviation of Rouge scores : Rouge R1 and Rouge R2 and R3

model	R-S1 / R	R-S1 / P	R-S1 / F1	R-S2 / R	R-S2 / P	R-S2 / F1
XPROPHETNET	0.392	0.403	0.366	0.385	0.400	0.361
MBARTHEZ	0.364	0.387	0.347	0.357	0.383	0.342
RULEBASED	0.374	0.382	0.346	0.372	0.379	0.343
BARTHEZ	0.363	0.381	0.340	0.355	0.377	0.335
BYT5	0.334	0.337	0.307	0.327	0.334	0.302
MT5	0.315	0.329	0.296	0.307	0.324	0.290
FR-RL-SUMM	0.318	0.327	0.294	0.309	0.316	0.284
LEAD-200	0.302	0.299	0.273	0.294	0.295	0.268
FIRSTPARAGRAPH	0.301	0.335	0.269	0.297	0.336	0.266
GRAPH-ROBERTA	0.386	0.163	0.201	0.372	0.162	0.198
BERT-CLUSTERING	0.143	0.309	0.171	0.137	0.299	0.164
GRAPH-DISTILIUSE	0.325	0.139	0.170	0.311	0.136	0.166
TEXTRANK	0.163	0.185	0.158	0.153	0.176	0.149
LEXRANK	0.265	0.125	0.146	0.251	0.122	0.141
MUSE	0.183	0.145	0.143	0.171	0.135	0.134
POLY	0.148	0.156	0.134	0.137	0.146	0.124
WORDFREQUENCY	0.469	0.096	0.132	0.453	0.092	0.127
LEXRANK-CONTINUOUS	0.056	0.202	0.066	0.049	0.186	0.059

Table 8.4: Reporting the Rouge scores : Rouge S1 and Rouge S2

model	R-S1 / R	R-S1 / P	R-S1 / F1	R-S2 / R	R-S2 / P	R-S2 / F1
XPROPHETNET	0.31	0.32	0.28	0.31	0.33	0.28
MBARTHEZ	0.31	0.32	0.28	0.31	0.33	0.28
RULEBASED	0.39	0.34	0.36	0.39	0.34	0.36
BARTHEZ	0.30	0.32	0.27	0.30	0.33	0.27
BYT5	0.30	0.29	0.26	0.30	0.30	0.26
MT5	0.28	0.29	0.25	0.28	0.29	0.26
FR-RL-SUMM	0.23	0.28	0.22	0.23	0.27	0.21
LEAD-200	0.28	0.26	0.23	0.28	0.27	0.24
FIRSTPARAGRAPH	0.38	0.31	0.33	0.38	0.31	0.33
GRAPH-ROBERTA	0.27	0.15	0.16	0.27	0.15	0.16
BERT-CLUSTERING	0.20	0.27	0.19	0.20	0.27	0.19
GRAPH-DISTILIUSE	0.25	0.14	0.14	0.25	0.14	0.14
TEXTRANK	0.14	0.19	0.13	0.14	0.19	0.13
LEXRANK	0.23	0.13	0.13	0.23	0.13	0.13
MUSE	0.19	0.17	0.15	0.19	0.17	0.15
POLY	0.13	0.13	0.09	0.13	0.13	0.09
WORDFREQUENCY-ALGO	0.18	0.12	0.11	0.18	0.11	0.11
LEXRANK-CONTINUOUS	0.07	0.22	0.07	0.07	0.21	0.07

Table 8.5: Reporting the Standard deviation of Rouge scores: Rouge S1 and Rouge S2

Table 8.6 presents the Rouge L and Rouge SU4 scores for various summarisation models. The results indicate that XPROPHETNET is the highest-performing model in terms of Rouge L and Rouge SU4 F1 scores, with scores of 0.466 and 0.386, respectively. MBARTHEZ and BARTHEZ follow closely behind, with Rouge L recall scores of 0.451 and 0.446, respectively. Conversely, RULEBASED performs less well compared to the previous rouge variants. Regarding the weakest-performing models, LEXRANK-CONTINUOUS and word frequency algorithms achieve the lowest Rouge-L and Rouge Su4 scores. Overall, the tables provide insight into the relative strengths and weaknesses of various summarisation models regarding several Rouge variants. XPROPHETNET is the top-performing model overall.

model	R-L / R	R-L / P	R-L / F1	R-SU4 / R	R-SU4 / P	R-SU4 / F1
XPROPHETNET	0.478	0.500	0.466	0.412	0.426	0.386
MBARTHEZ	0.461	0.484	0.451	0.385	0.411	0.368
BARTHEZ	0.461	0.478	0.446	0.383	0.405	0.361
BYT5	0.428	0.448	0.415	0.356	0.364	0.330
MT5	0.414	0.443	0.407	0.337	0.355	0.319
RULEBASED	0.408	0.461	0.398	0.383	0.454	0.356
FR-RL-SUMM	0.416	0.423	0.397	0.343	0.338	0.309
LEAD-200	0.395	0.421	0.386	0.323	0.329	0.296
GRAPH-ROBERTA	0.508	0.290	0.343	0.416	0.180	0.221
FIRSTPARAGRAPH	0.345	0.463	0.340	0.310	0.395	0.281
GRAPH-DISTILIUSE	0.443	0.282	0.320	0.359	0.156	0.191
LEXRANK	0.385	0.265	0.287	0.299	0.146	0.169
TEXTRANK	0.264	0.272	0.268	0.197	0.216	0.188
BERT-CLUSTERING	0.213	0.439	0.265	0.153	0.350	0.187
POLY	0.272	0.285	0.262	0.177	0.190	0.162
MUSE	0.274	0.280	0.258	0.213	0.163	0.163
WORDFREQUENCY-ALGO	0.586	0.169	0.232	0.509	0.102	0.141
LEXRANK-CONTINUOUS	0.116	0.385	0.152	0.067	0.248	0.080

Table 8.6: Reporting the Rouge scores: Rouge L and Rouge Su4

model	R-L / R	R-L / P	R-L / F1	R-SU4 / R	R-SU4 / P	R-SU4 / F1
XPROPHETNET	0.27	0.28	0.26	0.30	0.31	0.27
MBARTHEZ	0.27	0.28	0.25	0.29	0.31	0.27
BARTHEZ	0.27	0.28	0.24	0.29	0.31	0.26
BYT5	0.27	0.28	0.23	0.29	0.28	0.25
MT5	0.25	0.28	0.23	0.26	0.28	0.24
RULEBASED	0.39	0.28	0.36	0.39	0.32	0.36
FR-RL-SUMM	0.21	0.28	0.21	0.21	0.26	0.20
LEAD-200	0.25	0.28	0.22	0.27	0.26	0.23
GRAPH-ROBERTA	0.22	0.28	0.16	0.25	0.15	0.15
FIRSTPARAGRAPH	0.37	0.28	0.32	0.38	0.28	0.33
GRAPH-DISTILIUSE	0.21	0.28	0.15	0.24	0.14	0.14
LEXRANK	0.21	0.28	0.13	0.22	0.13	0.13
TEXTRANK	0.13	0.28	0.13	0.13	0.18	0.13
BERT-CLUSTERING	0.19	0.28	0.18	0.19	0.25	0.18
POLY	0.13	0.28	0.10	0.12	0.13	0.09
MUSE	0.18	0.28	0.14	0.18	0.16	0.14
WORDFREQUENCY-ALGO	0.17	0.28	0.12	0.17	0.12	0.12
LEXRANK-CONTINUOUS	0.10	0.28	0.09	0.08	0.21	0.07

Table 8.7: Reporting the standard deviation of Rouge L and Rouge Su4

Overall, we can confirm an overperformance of supervised models over unsupervised models in this case of the French financial summarisation task. However, with the advances that could happen in dataset labelling, we can, in the near future extract very accurate summaries using only unsupervised models.

8.10.4 Pairwise Statistical Testing

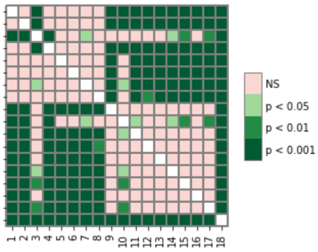
In this section, we will carry out pairwise comparisons between all systems to assess whether system differences are statistically significant.

Different p-value adjusting: Figure 8.6 shows different significance plots. We plot a heatmap of p values using Seaborn (Waskom et al., 2017) with different adjusting methods for calculating p-value for the Rouge 2 F1 score using the Conover pairwise test. We used 'bonferroni' (one-step correction), 'sidak' (one-step correction), 'holm-sidak' (step-down method using Sidak adjustments), 'holm' (step-down method using Bonferroni adjustments), 'simes-hochberg' (step-up method (independent)), 'hommel' (closed method based on Simes tests (non-negative)), 'fdr_bh' (Benjamini/Hochberg (non-negative)), 'fdr_by' (Benjamini/Yekutieli (negative)), 'fdr_tsbh' (two stage fdr correction (non-negative)) and 'fdr_tsbky'(two stage fdr correction (non-negative)).

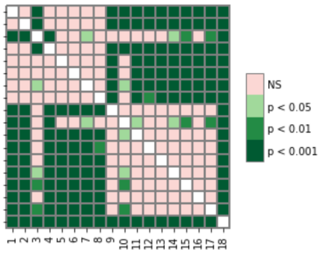
Then, we performed different statistical tests for several rouge scores, and we used the Holm adjusting method for all our calculations. We performed several statistical tests as stated below:

- posthoc_conover: Post hoc pairwise test for multiple comparisons of mean rank sums (Conover´s test). It may be used after Kruskal-Wallis one-way analysis of variance by ranks to do pairwise comparisons (Conover et al., 1979).
- posthoc_dunn : Post hoc pairwise test for multiple comparisons of mean rank sums (Dunn's test). May be used after Kruskal-Wallis one-way analysis of variance by ranks to make pairwise comparisons (Dunn, 1964)
- posthoc_mannwhitney: Pairwise comparisons with Mann-Whitney rank test.
- posthoc_nemenyi: Post hoc pairwise test for multiple comparisons of mean rank sums (Nemenyi's test). It may be used after Kruskal-Wallis one-way analysis of variance by ranks to make pairwise comparisons.
- posthoc_ttest: Pairwise T-test for multiple comparisons of independent groups. May be used after a parametric ANOVA to do pairwise comparisons[21],[22].
- posthoc_vanwaerden: Van der Waerden's test for pairwise multiple comparisons between group levels[23]. (Elamir, 2022; Conover et al., 1979)

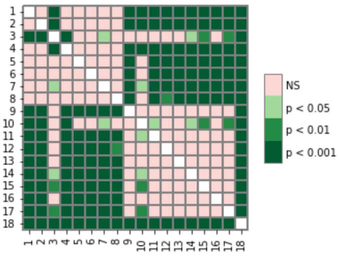
[21] http://en.wikipedia.org/wiki/T-test#Independent_two-sample_t-test
[22] http://en.wikipedia.org/wiki/Welch%27s_t_test
[23] https://en.wikipedia.org/wiki/Van_der_Waerden_test



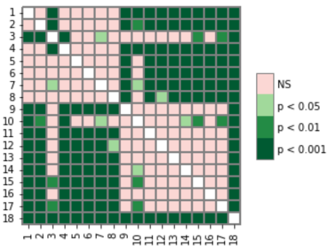
(a) Holm



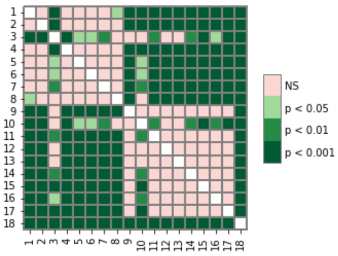
(b) holm sidak



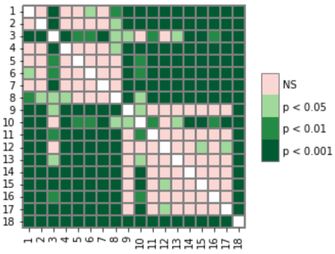
(c) hommel



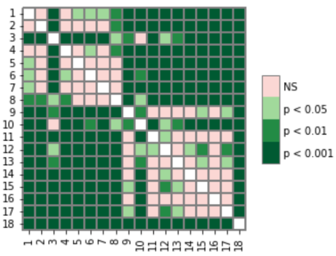
(d) sidak



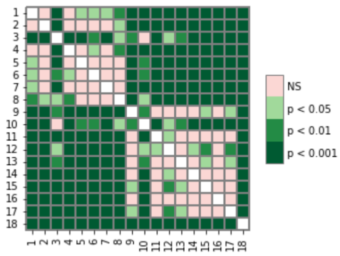
(e) fdr_by



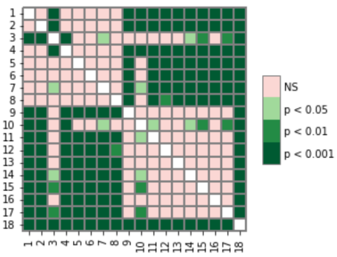
(f) fdr_bh



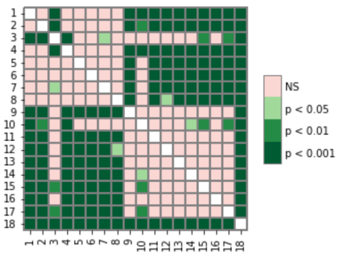
(g) fdr_tsbh



(h) fdr_tsbky

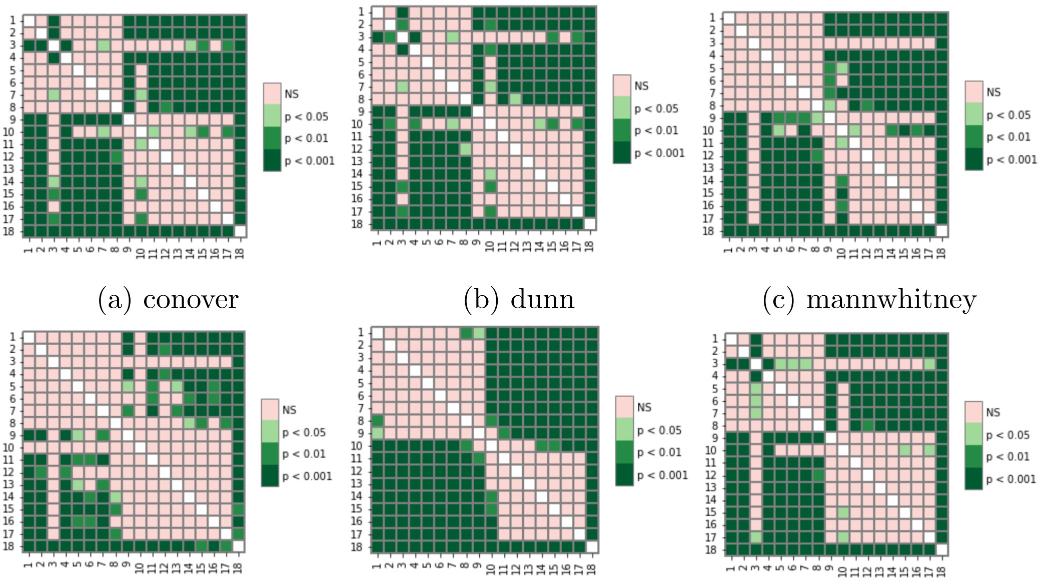


(i) simes-hochberg



(j) bonferroni

Figure 8.6: Different Adjusting methods for calculating p-value for the Rouge2 F1 score using the conover pairwise test.

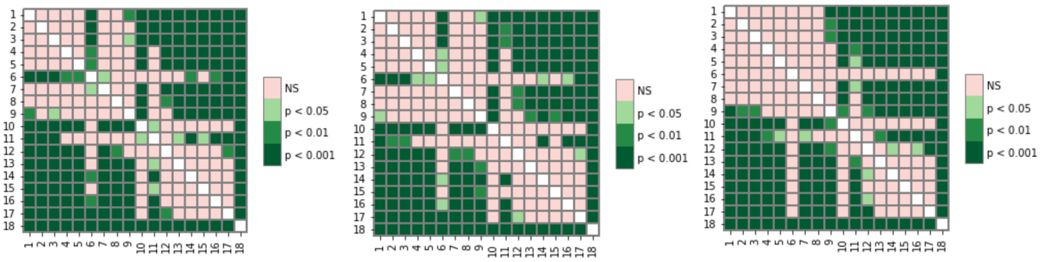


(a) conover (b) dunn (c) mannwhitney

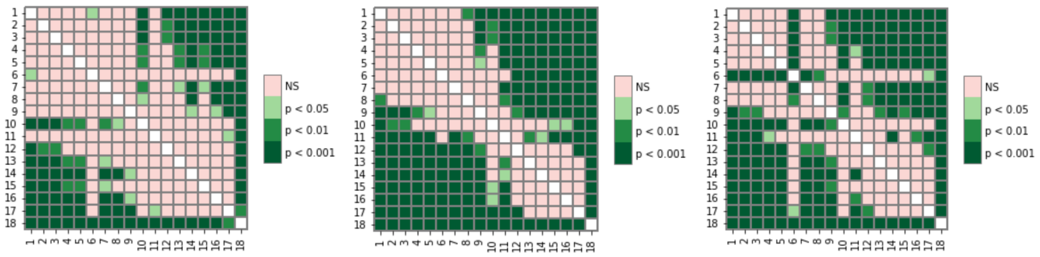
(d) nemenyi (e) ttest (f) vanwaerden

Figure 8.7: Different Pairwise Statistical Tests applied on the Rouge2 of all the summarisation systems: the numbers ranging from 1 to 18 represent the systems ordered by R2. 1 will be Xprophetnet, and 18 will be Lexrank continuous.

The figures 8.7, 8.8 and 8.9 display the heatmaps representing p-values for Rouge 2, Rouge R-L and Rouge SU4. Each cell in the heatmap corresponds to the significance level of a comparison test between two systems. The cell(row2, column1) shows the colour of the result of the comparison test between system 1 and system 2. The colour intensity indicates the level of significance, with darker shades representing higher significance. Notably, the heatmaps are symmetric, and the most substantial significance change occurs when transitioning from supervised to unsupervised models.

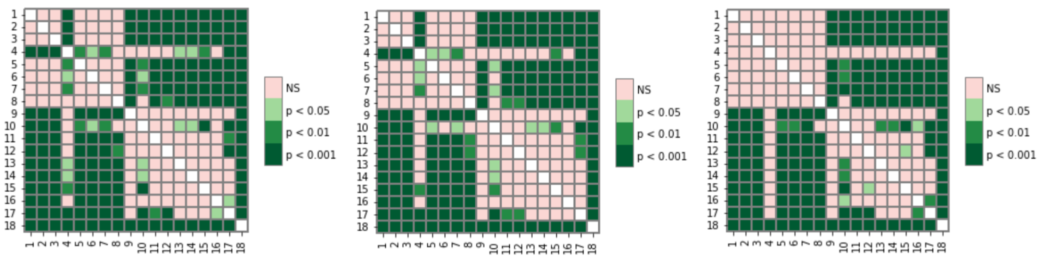


(a) conover (b) dunn (c) mannwhitney

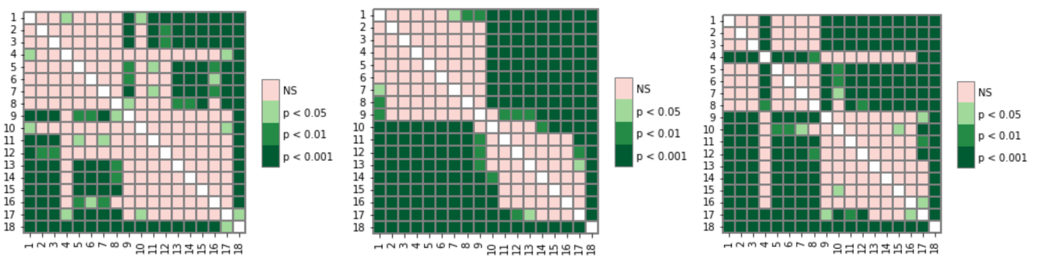


(d) nemenyi (e) ttest (f) vanwaerden

Figure 8.8: Different Pairwise Statistical Tests applied on the Rouge-L of all the summarisation systems: the numbers ranging from 1 to 18 represent the systems ordered by r L. 1 will be Xprophetnet, and 18 will be Lexrank continuous.



(a) conover (b) dunn (c) mannwhitney



(d) nemenyi (e) ttest (f) vanwaerden

Figure 8.9: Different Pairwise Statistical Tests applied on the Rouge SU4 of all the summarisation systems: the numbers ranging from 1 to 18 represent the systems ordered by R SU4. 1 will be Xprophetnet and 18 will be Lexrank continuous.

8.11 Summary of the Chapter

The chapter presented a monitoring study within French narrative summarisation. This study monitors the evolution of evaluation metrics during the French financial narrative summarisation task. It was performed on the CoFiF Plus dataset and benchmarked four summarisation techniques (monolingual seq2seq models, multilingual seq2seq models, Encoder2encoder models, and encoder2decoder models). The study evaluates the performance of different summarisation models using a range of metrics. It monitors n-gram-based (rouge 1,2, L, Lsum), embeddings-based (Frugal score, Bleurt, Bert score) and statistical-based metrics (Depth score, Bary score and infoLM score). This chapter provides a proof of concept that we can summarise using an encoder-only or a decoder-only model without needing to pre-train monolingual seq2seq models, which are very costly to train. However, this study proves that encoder-only summarisers are weak learners compared to native encoder-decoder models based on output from different metrics. Also, It shows that the model-based metrics converge and improve faster than the statistical-based metrics or n-gram-based metrics. Furthermore, this chapter explored different French summarisation techniques by developing different unsupervised and heuristic rule-based summarisation algorithms for French financial statement summarisation. These experiments show the robustness of unsupervised techniques to extract the essential parts in an annual report, especially if the dataset is well annotated. This opens the doors for more unsupervised summarisation with the introduction of the new reporting framework in France using the XBRL markup language. Overall, our findings provide valuable insights for researchers working on French financial narrative summarisation and highlight the importance of carefully selecting and monitoring evaluation metrics. In future work, we plan to extend our approach to other languages and investigate the potential of using additional evaluation metrics to further improve the performance of summarisation systems. We hope that our work will contribute to the development of more accurate and efficient summarisation systems for a variety of applications.

Acknowledgements: We would like to acknowledge the support of the Weights and Biases (wandb.ai) MLOps platform (Biewald, 2020), which provided us with an excellent platform for running experiments and tracking their results. The platform's features helped us synchronise our transformer summarisation workflow on the Lancaster University HEC cluster with the wandb cloud server. We are grateful for the free access offered to academics, which has contributed significantly to the success of our research.

Chapter 9

Conclusion

9.1 Thesis Summary and Concluding Remarks

In this chapter, we conclude the dissertation by providing a concise overview of the main contributions, which have been discussed in the preceding chapters. Additionally, we will highlight various promising directions for future work that require further investigation. This chapter will wrap up the various topics discussed in the thesis. It will start by providing a summary of each chapter. Following that, we will revisit the research questions posed in Chapter 1. The next step involves presenting the future work. Lastly, we will introduce a new framework to prepare financial narrative summarisation datasets.

The thesis explored three languages: English, Arabic and French, with different research tasks performed on every language with different input document sizes and summarisation techniques. This required different experiments, including data creation, data labelling, model training and fine-tuning, and model evaluation through quantitative methods, statistical evaluation and adversarial attacks.

The work on English focuses on very long documents and explores more summarisation techniques, transformers, financial embeddings, evaluation metrics, validation techniques, statistical techniques and result analysis. The Arabic work focuses on financial news, which are short documents and highly abstractive. The Arabic-focused part of the thesis introduces a new pretrained model and comes to solving the issue of unavailable financial pretrained models in Arabic. In addition, this work aims to explain how we can use transformers for generative tasks (abstractive summarisation) in order to summarise financial news generated by news agencies in Arabic. The French work introduced a different aspect of NLP research work, which is corpus creation. This thesis presents a new financial summarisation corpus composed of medium size documents. In addition, the thesis introduced the first training monitoring study within French narrative summarisation.

This thesis, in chapter 2, summarised the literature and previous work related to text summarisation, natural language processing, financial narrative processing, transformers, Seq2seq models and automatic evaluation. In chapter 3, we outlined the methodology employed in the design of the research process. In chapter 4, we introduced how to train a custom field financial Word Embeddings from scratch. In chapter 5, we introduced different financial corpus in three Languages. We gave different statistics and plots about every corpus. These datasets were used in the next chapters. In chapter 6, we addressed the task of long document financial extractive summarisation in English. In chapter 7, we addressed abstractive summarisation on Arabic financial news summarisation. In chapter 8, we addressed the monitoring of training French model summarisers on medium size financial document. The dataset was collected from publicly listed French companies.

9.2 Research Questions Revisited

In this section, we will revisit the research questions posed in chapter 1 and look at how the work completed has answered them. We will go through the research questions and try to explicitly explain our findings. We will restate each question, summarise the key findings, discuss their implications, and suggest directions for future research.

- **How can we enhance financial report summarisation? (RQ1)**
  - 1.1 What are the most effective methods for financial annual report summarisation, and how do they vary for different narrative types?
  - 1.2 What methods are suitable for summarising annual reports (long documents) and financial news (short documents)?
  - 1.3 How can pre-trained language models be adapted for very long documents, taking into consideration the memory limitations of many Large Language Models for handling very long documents?
  - 1.4 What are the specific aspects of financial text that affect the quality of text summarisation and that make it different from other types of documents (e.g. legal, healthcare)?

  This research question has been mainly addressed in Chapter 6 and 8 where we highlighted different techniques of summarisation (Supervised and unsupervised) and we explained how we can deal with long financial reports. We demonstrated that we can adapt large language models to the task of annual report summarisation by transforming the problem of summarisation into a task of predicting the start of a good summary candidate. This is very powerful in the case where we have a block based summarisation dataset. Furthermore, we showed that rule-based and unsupervised methodologies generate competitive results, especially in chapter 8. Moreover, we showed that we can summarise using reinforcement learning-based models, but we were unable to have high scores.

In chapter 7, we explained how we can summarise financial news using monolingual text-to-text models by leveraging the power of Transformer encoder-decoder architecture to generate fluent text. Our analysis revealed that using monolingual models would enhance the readability of the generated text.

In chapter 2, we explained the specific aspects of financial text compared to other types of documents. We described how financial documents come with a special tone and a special set of vocabulary which requires special models and special preprocessing techniques.

While the results are promising, further research is needed to explore more recently released large and tiny language models with bigger memory capacities.

- **How do we improve evaluation methods for text summarisation? (RQ2)**
  - 2.1 What other NLP methods can we exploit to better model text similarity in order to improve the evaluation process?
  - 2.2 What can we learn about the different categories of evaluation metrics by testing their behaviour during the training process?
  - 2.3 How can we confirm quantitative NLP results by human evaluation and readability measure?

In chapter 6, we explored different evaluation measures, such as ngram matching and embedding based ones and performed data visualisation for our result distributions. We showed that ngram-based metrics are still a good benchmark for extractive summarisation evaluation. This was done through different adversarial attacks to measure the robustness of our used metrics. Furthermore, we measured the correlation between different metrics in chapter 6. Our experiments revealed that different variants of the same metrics are highly correlated and that different categories of metrics are less correlated.

In chapter 8, we introduced an MLOps study where we monitored the behaviour of different evaluation metrics from different categories. We proved that some metrics such as Infolm, depthscore or Cider gave random behaviour and we can not rely on them for evaluating NLP systems. In addition, we showed that ngram and embedding-based metrics still to be the reference in such extractive summarisation use cases.

In chapter 7, we explored using the Arabic readability score as a robust way to evaluate generated text. Furthermore, we investigated the use of human evaluation which is a very strong natural language generation evaluation tool. We adopted a best-worst scaling methodology. Although human evaluation is more accurate than quantitative techniques, it requires a high budget in order to have a significant result that we can generalise.

- **How can we move beyond the current focus only on the English language for financial text summarisation? (RQ3)**
  - 3.1 How can we enhance multilingual financial NLP research?
  - 3.2 How portable are multilingual financial text summarisation methods from English to other languages?
  - 3.3 How can we boost financial NLP research in under-researched languages by pertaining and finetuning field-specific language models?

  In chapter 7 and chapter 8, we explored Arabic and French languages as less covered languages, and we showed that the same techniques used in English can be portable to French. However, some models pretrained for English are not useful for other languages. We need either multilingual models or monolingual-specific financial models adapted to the specificities of each language. In addition, we pretrained word embedding models in different languages in chapter 4. We also created a French summarisation dataset in chapter 5 in order to have a similar dataset to the summarisation English corpus that we already have and presented in chapter 5. We showed that a well-labelled dataset may improve the quality of the generated summaries and make unsupervised techniques more suitable. Future research work may be focused on exploring other languages, creating other financial corpora and improving the labelling process.

9.3 Research Limitations

The field of Natural Language Processing (NLP) encounters several significant limitations. Generally speaking, the biggest limitation in NLP research is the lack of qualitative data of high quality especially for low-resource languages (Kuchmiichuk, 2023). Although the amount of data available online has increased drastically during the last few years, there is still a lack of quality data suitable for advanced NLP use cases. Hence, one of our top research objectives is to develop more Multilingual Financial NLP resources and tools. This will enable us to continue the pretraining of large language models useful for different tasks. In addition, there is another limitation, which is finding an optimal evaluation method for generated summaries. Although human evaluation is the standard in summary evaluation, its expensive and time-consuming nature make it impractical for large-scale evaluations (Srinivasan et al., 2021), leading to difficulties in achieving statistically significant results. Furthermore, computational memory stands as the bottleneck of NLP research (Wang et al., 2023). The constraints in computational resources limit the scale and efficiency of NLP experiments and model training. The main problem that we encountered as NLP researchers was computational power and access to GPU and TPU resources. Finally, we see that exploring languages beyond English presents another hurdle as it often requires specific models and specific datasets tailored to the linguistic characteristics of each language (Fan et al., 2021) since the ability of multilingual models may sometimes be limited. In conclusion, addressing the limitations in NLP research, such as the scarcity of high-quality data, computational constraints, evaluation methodologies, and language diversity, is crucial for advancing the field and developing more robust multilingual NLP resources and tools.

9.4 Future Work

The new knowledge uncovered in the thesis presents numerous opportunities for additional research in the field of financial text summarisation. This section will prompt further directions which can guide future financial summarisation research. Furthermore, Saggion et al. (2012) presented a survey entitled "Automatic Text Summarisation: Past, Present and Future" which states the future direction in the field of text summarisation in general. In the case of multilingual financial text summarisation, the future work could be defined as follows:

- Pretraining from scratch a large financial decoder-only model like bloombergGPT (Wu et al., 2023).
- Continue the hard work on corpora collection and labelling: In this thesis, we have shown the importance of creating datasets in different languages. Future work should look for new sources of data in a multilingual context.
- Explore more evaluation methods going beyond quantitative automatic metrics.
- Improving the result of summarisation by using newly introduced large language models and newly MLOps and advanced hyperparameter search techniques.
- Work on building new abstractive summarisation corpora for English and French to substitute the current highly extractive ones.
- Exploring more languages such as Spanish, Portuguese and German.

9.5 ChatGPT: Future of Dataset Creation: An ultimate guide on producing a human-labelled financial narrative summarisation dataset

As we previously discussed in chapters 6 and 8, having sufficient data relevant to the finance domain is crucial. The major obstacle to utilising NLP approaches for summarising financial documents is the lack of gold standards for these documents and their associated summaries.

This section provides a comprehensive guide on creating a human-labelled financial narrative summarisation dataset in the ChatGPT era with a case study focusing on

UK-listed companies. We have a corpus of financial documents from UK-listed firms ranging from 1998 to 2018. The compiled data set produces around 4000 reports that could be used in training material for NLP systems, making it possible for them to describe financial stories precisely. Our work contributes to financial NLP research, providing a blueprint for creating high-quality labelled datasets that can speed up the development of automated financial analysis tools.

9.5.1 Proposed Frameworks

ChatGPT[1] is an OpenAI-developed language model that can produce responses that resemble a human's when given text-based prompts. It utilises the transformer architecture, which allows it to infer the meaning of input sentences and generate appropriate answers. This model has been trained with vast amounts of data from various sources, giving it a wide range of understanding across many topics. It can respond to inquiries, take part in conversations, compose essays and even create literary works such as poetry and stories. ChatGPT can be used in many different ways, such as providing customer service, translating languages, and generating content. It is available to the public through various API interfaces, allowing developers to incorporate it into their programs and services. ChatGPT has not been tailored to the financial field, but it was trained on a large assortment of text materials, including financial news and reports. Therefore, it possesses a certain degree of knowledge of financial terms and ideas and can provide pertinent information or observations concerning finance. Nonetheless, it should be remembered that ChatGPT is an AI language model rather than a financial analyst.

Here, we suggest two frameworks for users to create a financial narrative summarisation corpus for UK-listed companies.

9.5.1.1 Framework 1:

The first framework is a chatGPT-aided solution to create a gold summary. We will work on the case of HSBC's financial annual report for the tax year 2019[2].

The obvious prompt that would come to your mind when dealing with chatgpt asking for help in this task is to directly ask the bot: could you give me a summary of the HSBC group annual report for the tax year 2019? In our case, we avoid this method because chatgpt will generate a very narrative and short summary that will not include the key financial numbers that company stakeholders are interested in. Using this approach will not enhance our dataset labelling process. In contrast, we will generate the gold standard summary by asking different questions and then organise and combine them to get coherent sentences. We tried different prompts and finally came out with the following four prompts to help construct a consistent summary.

[1] `https://chat.openai.com/chat`

[2] `https://www.annualreports.com/HostedData/AnnualReportArchive/h/NYSE_HSBC_2019.pdf`

- Could you give me a summary of the HSBC group chairman highlights in the annual report for the tax year 2019?
- Could you give me a summary of the financial highlights of the HSBC group during the tax year 2019?
- Could you give me a summary of the financial statements of HSBC group during the tax year 2019?
- Could you give me a summary of the HSBC group's future perspectives in the annual report for the tax year 2019?

Once we get the responses, we can coherently combine them. We performed this exercise ourselves, and we obtained an example gold summary. Table 9.1 includes the proposed summary for HSBC's annual report regarding the tax year of 2019. the length of the summary is 600 words.

Disclaimer: We asked chatgpt: could you give me a summary of the financial highlights of HSBC group during the tax year 2021? I'm sorry, but I am unable to provide you with information about the financial performance of the HSBC Group in 2021. My training data only goes up until 2021, and I do not have access to real-time data or the ability to browse the internet. However, I can try to help you with any other questions you may have. Is there anything else I can assist you with?

Hence, we cannot use chatGPT to generate summaries for the recent financial annual reports because its training data is not incremental and especially since it is not a search engine and could not browse the internet.

Instead, by using this technique, researchers could create an abstractive summarisation corpus in a quick and efficient way. This will be a new milestone in our research topic. Hence we will replace the dataset that we are currently using, which is highly extractive. By getting such abstractive datasets, we can leverage the pretrained Longformer Encoder Decoder transformer to pre-train end-to-end financial summarisation abstractive models.

The chairman of HSBC Group is the head of the board of directors and plays a vital role in the overall governance and strategic direction of the company. In the annual report for the tax year ending December 31, 2019, the chairman of HSBC Group, Mark Tucker, highlighted several key achievements and challenges facing the company.
Some key highlights from the annual report include:

- The successful completion of a multi-year restructuring plan, which included the sale of underperforming businesses and the simplification of the group's structure.
- Strong growth in the group's retail banking and wealth management businesses, particularly in Asia.
- The group's continued focus on expanding its commercial banking business, with a particular emphasis on growth in China and other emerging markets.
- The group's efforts to improve its risk management and compliance processes, including the implementation of a new risk management framework.
- The group's commitment to sustainability and responsible business practices, including the achievement of gender balance in senior management roles.

HSBC Holdings plc is a British multinational investment bank and financial services company headquartered in London, England. The company is one of the largest banking and financial services organisations in the world. Here is a summary of some of the financial highlights of HSBC Group for the tax year ending December 31, 2019:
The financial statements are a key part of a company's annual report and provide detailed information about the company's financial position, performance, and cash flow. Here is a summary of the financial statements of HSBC Group for the tax year ending December 31, 2019:
Balance sheet:
Total assets: $2.7 trillion Total liabilities: $2.4 trillion
Income statement:
Revenue: $53.5 billion Net income: $10.3 billion
Statement of cash flows:
Net cash from operating activities: $18.4 billion Net cash used in investing activities: $4.9 billion Net cash from financing activities: $7.1 billion
The balance sheet shows the company's assets and liabilities at the end of the fiscal year, while the income statement shows the company's revenues, expenses, and net income for the year. The statement of cash flows shows the company's cash inflows and outflows from operating, investing, and financing activities.
Some key highlights for the year included:

- Improved performance in the group's retail banking and wealth management businesses, particularly in Asia.
- Continued growth in the group's commercial banking business.
- A significant increase in customer deposits, driven by strong growth in the group's retail banking and wealth management businesses.
- The successful completion of several strategic initiatives, including the sale of the group's operations in Turkey and the acquisition of a controlling stake in Ping An Good Doctor, a leading online healthcare platform in China.

In the annual report for the tax year ending December 31, 2019, HSBC Group outlined its future perspectives and strategic priorities for the coming years. Some key highlights from the annual report include:

- The group's focus on driving growth in its core businesses, particularly retail banking and wealth management, and on expanding its commercial banking business in key markets around the world.
- The group's commitment to delivering strong financial performance and shareholder value, through a focus on efficiency and cost management.
- The group's efforts to build a more sustainable and responsible business, through initiatives such as the reduction of carbon emissions and the promotion of diversity and inclusion.
- The group's plans to continue investing in technology and digital capabilities, to enhance the customer experience and improve operational efficiency.
- The group's intention to maintain a strong and well-capitalized balance sheet, through the management of capital and liquidity in a responsible and conservative manner.

Table 9.1: HSBC 2019 annual report proposed gold summary generated with the help of ChatGPT

9.5.1.2 Framework 2:

The second proposed methodology consists of manually extracting from the report and paraphrasing using one of the leading commercial solutions or with a finetuned text-to-text model on a downstream task of sentence paraphrasing. This idea is more difficult to achieve and needs much more time, but it is derived from the same idea that we are currently using to prepare an extractive dataset. In our financial narrative summarisation dataset, we use rule-based scripts that extract different sections such as the chairman highlights, the financial statements overview, the future perspectives and the external auditors report. Hence we will get a dataset labelled by different gold standards (3 on average). What if we extract these sections, paraphrase them (rewrite them) in a shorter way and combine all of them in one gold summary. Therefore we will get a very significant abstractive gold standard summary for our financial annual report. This technique becomes feasible with the emergence of several commercial SaaS solutions that provide such services. We can name few examples : Writesonic[3] , Heyfriday[4], quillbot[5], textcortex[6]. In addition, there are some open-source finetuned transformer models on the huggingface cloud repository[7]. But these models could have a limited text input length. That is why we advise using commercial solutions.

Table 9.2 shows an example of a gold standard we generated using our second method. We started by manually selecting the parts to use in a gold standard summary. We extracted the chairman's message, the CEO highlights, the financial statement overview, the financial highlights section and the future perspectives section. Then we rewrote them using a premium Writesonic account, and we arranged them to get a coherent gold standard summary.

[3] https://www.writesonic.com

[4] https://www.heyfriday.ai/

[5] https://www.quillbot.com//

[6] https://www.textcortex.com

[7] https://huggingface.co/mrm8488/t5-small-finetuned-quora-for-paraphrasing?

Group Chairman's statement: In our interim results, I noted that the external environment was becoming more intricate and demanding. The 2019 figures demonstrate this has been the case. Our reported profit before tax dropped by 33% due to the impairment of historical goodwill. Nevertheless, the robustness of our business model resulted in an adjusted pre-tax profit of $22.2bn - a 5% increase. Retail Banking and Wealth Management, Global Private Banking and Commercial Banking all performed well, whilst our top-notch transaction banking operation again showed our global network's success. This, combined with the Group's solid capital base, has allowed the Board to retain a dividend of $0.51 for 2019 without alteration.
Group Chief Executive's review: Our Group's financial performance showed a decrease in profit before tax of 33% compared to the previous year due to a $7.3bn goodwill impairment. This was caused by an alteration in long-term economic growth prospects and Global Banking and Markets restructuring. Adjusted profits before tax increased by 5%, with revenue growth seen across three out of four global businesses. Cost management was kept under control resulting in positive adjusted jaws of 3.1%. The Group's return on average tangible equity decreased from 8.6% to 8.4%.
We achieved good revenue expansion in our areas of focus. Our businesses in Hong Kong and HSBC UK (our British ring-fenced bank) displayed remarkable strength to generate adjusted revenue growth of 7% and 3%, respectively, despite the turbulence experienced in both locales during 2019. Our operations in Mexico, India, the ASEAN region, and mainland China also performed admirably. The main regions that suffered underperformance were our activities in the US and our European non-ring-fenced bank; both saw a decrease in their revenues and profits before tax.
Retail Banking and Wealth Management had a successful year, achieving adjusted revenue growth of 9%. This can be attributed to investments in better customer service and expansion, which enabled us to win new customers, increase deposits, and expand lending operations in our main markets, with mortgage lending particularly strong in the UK and Hong Kong. Additionally, our Wealth business profited from advantageous market conditions impacting Insurance. Commercial Banking grew adjusted revenue by 6%, with all major products and regions seeing an increase. Investment into new platforms, digital capabilities and more lending enhanced our capacity to draw in new customers and take advantage of larger margins - this was especially noticeable in Global Liquidity & Cash Management plus Credit & Lending. The past year posed a challenge for Global Banking and Markets, with economic uncertainty driving down customer activity, especially in the US and Europe. However, due to good results from its transaction banking operations, adjusted revenue only dropped by 1% compared to 2018. Global Private Banking benefited from its close ties with other global businesses, bringing in $23 billion of fresh funds and raising adjusted revenue by 5%.

Financial Overview: In 2019, a reported pre-tax profit of $13.3bn was down 33%, including a $7.3bn impairment of goodwill. However, the adjusted pre-tax profit of $22.2bn increased by 5%. The business performed well, yet certain areas needed to meet expectations and had an unfavourable effect on returns. RBWM and CMB global businesses saw revenue growth, especially in Asia, while GPB attracted net new money worth $23bn over the year. Unfortunately, income in GB&M dropped relative to 2018 due to economic instability and spread compression that had a negative impact primarily on Global Markets and Global Banking in Europe. Credit losses and other impairments ('ECL') rose compared to the favourable 2018 and accounted for 0.27% of average gross customer loans. Operating expenses were carefully monitored, with the pace of growth in adjusted operating expenses lower than the previous year, while we kept investing. This enabled us to report positive adjusted jaws in 2019. Our average return on tangible equity (RoTE) for 2019 was 8.4%. Due to difficulties in the revenue domain and a more pessimistic outlook, we are no longer expecting to hit our goal of 11% RoTE in 2020. To tackle this issue, we intend to restructure businesses that are not producing satisfactory results so that resources can be redirected towards higher-returning ones, reduce our considerable expenses and slim down the organisation. Since the start of January 2020, the coronavirus outbreak has been causing economic distress in Hong Kong and mainland China which could affect performance this year.

Delivering our strategy: We have adjusted our business plan due to the altered macroeconomic environment and interest rate outlook. We are no longer anticipating reaching our 2020 return on average tangible equity ('RoTE') goal, as stated in our third quarter 2019 results. Despite a strong performance in Asia and the Middle East and with our market-leading transaction banking services globally, other areas of our business have yet to do as well. This is all taking place in an atmosphere of increasing interest rates, steady global economic growth, and moderate geopolitical risk; therefore, we must realign our priorities and financial objectives accordingly.
We intend to increase the profitability of our investments and optimise the Group's operations to generate greater returns and strengthen our growth potential. Our business update outlines how we will become a more streamlined, straightforward, and competitive Group with the aim of becoming the world's premier international bank.

2020 outlook: Since the beginning of 2020, the coronavirus crisis has caused considerable difficulties for our personnel, providers, and consumers, especially in China and Hong Kong. We appreciate the hardships this brings and have plans to help them through these difficult times. Depending on how this situation evolves, any economic slowdown could affect our predicted credit losses in China and Hong Kong. In addition, we may experience reduced revenue due to decreased lending activities and transaction volumes, as well as additional credit losses caused by disrupting customer supply chains over a more extended period.

Table 9.2: HSBC 2019 annual report proposed gold summary generated using paraphrasing technique

9.5.2 Summary

In conclusion, this section presents how we can use ChatGPT and generative AI to create abstractive human-generated novel gold standards for UK financial annual reports. Generative AI will certainly contribute to shaping the future of dataset creation and NLP research. The development of a comprehensive guide for producing a human-labelled financial narrative summarisation dataset underscores the importance of high-quality training data in enhancing the accuracy and efficacy of large AI models.

9.6 Future of Financial Summarisation

The SEC (Security Exchange Commission) mandates that publicly traded companies must submit their financial statements in a digital format using XBRL tagging system for financial annual reports. This machine-readable language facilitates the exchange and analysis of financial information. Investors and analysts can easily access and review XBRL-tagged financial reports from any public company. The XBRL system provides advantages to companies, such as a uniform method of displaying their financial reports. Additionally, investors and analysts benefit from this system since they have an orderly way to access and examine corporate financial data. The French regulator is about to implement the XBRL tagging system starting from 2023, which will greatly simplify the process of summarisation by allowing for easier extraction of report sections (Alphonse, 2023). This should also speed up extractive summarisation research. However, abstractive summarisation remains a wide-open field of study. In fact, in their book, Ramin et al. (2013) explored how business reporting can be improved, through the emergence of new reporting technologies under the IFRS, such as the use of XBRL tagging. Ramin et al. (2013) explains how XBRL can offer considerable benefits to people who produce and rely on business reports.

9.7 Ethical Considerations

Environmental footprint: the leading models in text summarisation and NLP have become larger and increasingly data-hungry since the development of the Transformer architecture. The NLP community is becoming more cognisant of the environmental impacts of training these models, which frequently require days of energy-intensive GPUs for a single training cycle (Strubell et al., 2019). For this reason we used the Weights & Biases compute tracker[8] to measure the GPU compute. In addition the HEC offers a detailed report about the GPU use after finishing the training. Added to that, we can use the MLCO2 machine learning emissions calculator[9] to evaluate our use of GPU in terms of CO2 emissions. This work should be responsible for

[8]Wandb compute tracker library: `https://wandb.ai/`
[9]`https://mlco2.github.io/impact/#compute`

considerable CO_2 emissions and this is becoming an increasingly problematic issue for the NLP community.

Legal issues with NLP corpora: as stated by the the Corpus Creation guide [10], copyright issues in corpus creation are complex and unavoidable. In fact, financial data raises several copyright issues. Although a lot of financial qualitative data (e.g. financial reports) is publicly available, we are not allowed to redistribute it in its original PDF format. That is why we opt for using data in a converted text format. Moreover data scraping could raise copyright issues with news websites. Other type of financial data may be premium and not accessible for researchers. It is mainly sold by data providers for financial institutions.

.

[10] https://www.lancaster.ac.uk/fass/projects/corpus/ZJU/xpapers/Xiao_corpus_creation.pdf

For more information about the data and models, please get in touch with the UCREL NLP Group https://ucrel.lancs.ac.uk/

Appendix A

Chapter 5 Appendix

A.1 FinAraSum Examples


```
{
    "article": "اقترحت المفوضية الأوروبية تغييرات على خطتها لحظر النفط الروسي في محاولة لكسب تأييد الدول المعارضة، بحسب وكالة \"رويترز\". وسيشمل الاقتراح المعدل فترة انتقالية مدتها ثلاثة أشهر قبل فرض حظر على خدمات الشحن لنقل النفط الروسي، وذلك بدلا من مهلة شهر واحد، كان قد تم اقتراحها سابقا. كما يشمل المقترح زيادة في الاستثمارات لتحديث البنية التحتية النفطية للتخفيف من تأثير العقوبات على الطاقة الروسية في دول الاتحاد، إضافة إلى منح مهلة لكل من هنغاريا وسلوفاكيا وجمهورية التشيك قبل إلزامها بحظر النفط الروسي. وكانت المفوضية الأوروبية، كشفت الأربعاء، عن حزمة عقوبات جديدة على روسيا تتضمن حظر واردات النفط الروسي بحلول نهاية العام واستبعاد بنوك روسية من نظام سويفت الدولي، كما اقترحت إطلاق حزمة مساعدة لتعافي أوكرانيا. وقالت رئيسة المفوضية أورزولا فون دير لاين، إن دول الاتحاد الأوروبي ستوقف استيراد النفط ومنتجات التكرير الروسية، وذلك ضمن حزمة سادسة مقترحة من العقوبات على موسكو بسبب الحرب على أوكرانيا.",
    "category": "economy",
    "date": "2022-05-06",
    "title": "تعديلات في مقترح الاتحاد الأوروبي للاستغناء عن النفط الروسي.. هذه أبرزها"
}
```


Figure A.1: Example 1 FinAraSum

{
"article": "أعلنت شركة الأنابيب السعودية \"أنابيب السعودية\"، اليوم الاثنين، عن فوزها بعقود لتوريد أنابيب صلب للبترول والغاز لشركة تينارس غلوبال سيرفيس بالأوروجواي بقيمة تقدر بنحو 139 مليون ريال., وقالت \"أنابيب السعودية\" في بيان على \"تداول السعودية\"، إنها ستورد طلبات شراء الأنابيب في غضون 4 أشهر., وتوقعت أن ينعكس الأثر المالي لهذه العقود في الربع الثالث والربع الرابع للسنة المالية 2022., كانت الشركة السعودية لأنابيب الصلب \"أنابيب السعودية\"، قد فازت في مارس الماضي، بعقود لتوريد أنابيب صلب للبترول والغاز لشركة تينارس غلوبال سيرفيس بالأوروغواي بقيمة تقدر بنحو 88.5 مليون ريال., وفي فبراير الماضي، وقعت الشركة السعودية لأنابيب الصلب \"أنابيب السعودية\"، عقداً مع أرامكو السعودية بقيمة نحو 80.6 مليون ريال.",
"category": "companies",
"date": "2022-05-16",
"title": "\"أنابيب السعودية\" تفوز بعقود توريد إلى الأوروجواي بــ 139 مليون ريال"
}

Figure A.2: Example 2 FinAraSum

{
"article": "حقق المؤشر نيكاي الياباني أكبر مكاسبه في أكثر من شهر اليوم الثلاثاء، إذ انحسرت المخاوف بشأن تأثير السلالة أوميكرون المتحورة من فيروس كورونا، مما شجع المستثمرين على الإقبال على شراء أسهم مثل أسهم مجموعة سوفت بنك والأسهم المرتبطة بالسفر., وارتفع المؤشر نيكاي 1.89% إلى 28455.60 نقطة، مسجلا أكبر مكاسبه بالنسبة المئوية منذ الأول من نوفمبر تشرين الثاني. وتقدم المؤشر توبكس الأوسع نطاقا 2.17%، إلى 1989.85 نقطة., اقتفت الأسهم اليابانية أيضا أثر الزيادة في وول ستريت، إذ رحب المستثمرون ببعض التصريحات الإيجابية من مسؤول أميركي كبير بشأن المتحور الأخير., وقفز سهم مجموعة سوفت بنك، المستثمر العالمي في شركات التكنولوجيا الذي تسبب في تراجع نيكاي في الجلسة السابقة، 7.94%، ليكون صاحب أكبر نسبة مكاسب على المؤشر., وارتفعت الأسهم المستفيدة من استئناف أنشطة اقتصادية، إذ قادت أسهم شركات الطيران ومشغلي خدمات السكك الحديدية المكاسب من بين 33 مؤشرا فرعيا للقطاعات في البورصة، بزيادة 4.17 %، و3.06 %، على الترتيب.",
"category": "financial-markets",
"date": "2021-12-07",
"title": "نيكاي يحقق أكبر قفزة في شهر مع انحسار مخاوف أوميكرون"
}

Figure A.3: Example 3 FinAraSum

A.2 French Dataset

A.2.1 French Financial Communications Stakeholders

Cliff (French Association of the Professionals of Financial Communications) was established in 1987 and is a stakeholder in the production of the "Cadre et Pratiques de Communication Financière" (Financial Communications: Framework and Practices), issued in both French and English and updated every year. In collaboration with SFAF ("Société française des analystes financiers": French Society of Financial Analysts), Cliff has published a financial communications charter ("Charte de la communication financière"). With IFA ("Institut Français des Administrateurs": French Institute of Directors), Cliff also produced a collection of best practices on "Relations between senior management and the board of directors with regard to financial communications issues". (Source: `https://cliff.asso.fr/fr/`)
XBRL(eXtensible Business Reporting Language): is a computer language, open and free of rights, which makes it possible to precisely identify each financial information, like the revolution constituted by the "barcode" in distribution. A royalty-free reporting language based on the XML standard, it reduces costs and makes the transmission of financial information more reliable.
iXBRL: provides a mechanism to embed XBRL codes in XHTML documents which allows you to combine the advantages of XBRL (tagged data) and the readable presentation of an annual report.
XBRL France: is a 1901 law association registered in France. Its members are companies, public or consular institutions or individuals working to develop the XBRL standard in France. XBRL France is a jurisdiction of the XBRL International consortium and is a member of the regional organisation (Source: `https://www.xbrlfrance.org/`).
Info Financière: is the French government financial reports directory where you can search for any financial report using the ISIN or company name or code LEI or issue date or type of information (source: `https://info-financiere.fr/pages/recherche`).
AMF (Autorité des marchés financiers): is the financial market authority, which is the French regulator of French financial markets.
French Index: The French index is called CAC40 (France's 40 largest market capitalisations). French firms are listed on the Euronext exchange, which includes firms from France, Netherlands, Belgium, and Portugal. The composition of CAC40 is updated quarterly by a committee of experts named the "Scientific Index Council".

A.2.2 Examples of gold standard summaries

This part will include examples of gold standard summaries extracted to create our summarisation dataset.

Report	(ENGIE 2018)
Gold Standard #1	Commentant les résultats annuels 2018, Isabelle Kocher, Directrice Générale d'ENGIE, a déclaré : Nous avons posé les jalons d'une importante création de valeur pour nos actionnaires et comptons sur nos réalisations pour être à l'avant-garde de la deuxième vague de la transition énergétique, avec un impact positif croissant sur nos clients. J'aimerais remercier tous les employés d'ENGIE pour leur engagement qui a été essentiel à la réalisation de notre plan stratégique au cours des trois dernières années. En 2018, nous avons atteint nos objectifs grâce à l'engagement de nos équipes, et ce malgré les défis exceptionnels que nous avons dû relever en Belgique.
Gold Standard #2	Atteinte des objectifs annuels : résultat net récurrent part du Groupe de 2,5 milliard d'euros, ratio dette nette / Ebitda à 2,3x. Stabilité de l'Ebitda qui démontre la solidité du modèle d'ENGIE, une dynamique sous-jacente positive des activités de croissance qui compense les impacts financiers défavorables dus aux importantes maintenances non programmées d'unités nucléaires en Belgique, à des effets de change négatifs et à l'effet dilutif des cessions. Croissance organique1 de l'Ebitda solide, à 5 %, qui reflète la progression des activités stratégiques du Groupe, particulièrement notable sur les activités Renouvelables et Solutions Clients BtoB et BtoT. Une réduction de la dette nette du Groupe (- 1,4 milliard d'euros vs. fin 2017) grâce à une robuste génération de cash opérationnelle2 et aux cessions. La solidité de la structure financière du Groupe est confirmée par les agences de notation qui placent le Groupe en tête de son secteur. Bilan du plan stratégique 2016-2018 : un portefeuille d'actifs reconfiguré, moins exposé aux prix de marché, moins carboné et présentant un potentiel de croissance amélioré. Une transformation permise par un programme de rotation de portefeuille (16,5 milliards d'euros3 de cessions quasiment finalisées), des investissements stratégiques (14,3 milliards d'euros4 d'investissements de croissance réalisés), des gains de performance (1,3 milliard d'euros de gains nets au niveau de l'Ebitda depuis 2015), le développement d'une force commerciale davantage orientée client ainsi que par l'accélération du développement dans les énergies renouvelables.
Gold Standard #3	Faits marquants opérationnels du Groupe depuis janvier 2018 Production d'électricité Renouvelable et Thermique contracté En France, le Groupe a confirmé sa position de N°1 dans le solaire et l'éolien en remportant 230 MW lors du dernier appel d'offres gouvernemental et par l'acquisition d'un portefeuille de projets de 1,8 GW (acquisition de LANGA, 1,3 GW ; acquisition de SAMEOLE, 500 MW). Par ailleurs, la société FEIH, détenue conjointement par ENGIE et Crédit Agricole Assurances, a atteint 1,5 GW de capacités solaires et éoliennes installées début 2019. Aux Etats-Unis, ENGIE a acquis Infinity Renewables et est ainsi devenu un leader dans le développement de parcs éoliens. La société a déjà développé 1,6 GW de capacités et possède un portefeuille de projets de 8 GW à divers stades de développement. En Inde, le Groupe a mis en service le parc solaire de Mirzapur et a atteint 1 GW de capacités renouvelables (éolien et solaire, installées ou en construction) en remportant un nouveau projet éolien de 200 MW. En Espagne, le Groupe a annoncé le développement de 9 parcs éoliens

Table A.1: ENGIE 2018 annual report gold standard summaries

Annual report	**(LVMH 2015)**
Gold Standard #1	M. Bernard Arnault, Président-Directeur Général de LVMH, a déclaré : Les résultats de 2015 confirment la capacité de LVMH à progresser et à gagner des parts de marché malgré le contexte d'instabilité économique et géopolitique. Les ventes et le résultat opérationnel atteignent un niveau record. Volonté d'excellence, obsession de la qualité, force d'innovation soutiennent notre dynamique, autant de valeurs illustrées par la Fondation Louis Vuitton et son bâtiment emblématique qui a accueilli plus d'un million de visiteurs en 2015. Toutes nos Maisons ont fait preuve en 2015 d'une grande réactivité. En adaptant leur stratégie aux évolutions du monde et en poursuivant leur développement, elles ont illustré la créativité et l'esprit d'entreprise qui les animent. Dans un contexte économique toujours incertain, nous pouvons compter sur la désirabilité de nos marques et l'agilité de nos équipes pour renforcer encore en 2016 notre avance dans l'univers des produits de haute qualité. L'année 2015 a été marquée par : • Des ventes et un résultat opérationnel courant records, • La forte progression en Europe, aux Etats-Unis et au Japon, • Un impact de change positif, • Une bonne performance des Vins et Spiritueux dans toutes les régions du monde avec une normalisation progressive de la situation en Chine, • Le succès des modèles iconiques et des nouveautés chez Louis Vuitton dont la rentabilité se maintient à un niveau exceptionnel, • La progression des marques de Mode, en particulier Fendi, Céline, Givenchy et Kenzo, • Une dynamique remarquable de Christian Dior qui gagne des parts de marché partout dans le monde, • Les excellents résultats de Bvlgari et le succès de la stratégie de recentrage de Tag Heuer, • La progression exceptionnelle de Sephora qui renforce ses positions dans tous ses marchés et dans le digital, • Un cash flow disponible de 3,7 milliards d'euros, en hausse de 30 %, • Un gearing qui s'établit à 16 % à fin décembre 2015.
Gold Standard #2	Excellente performance de LVMH en 2015 Ventes et Résultat opérationnel records Paris, le 2 février 2016 LVMH Moët Hennessy Louis Vuitton, leader mondial des produits de luxe, réalise en 2015 des ventes de 35,7 milliards d'euros, en progression de 16 %. La croissance organique des ventes ressort à 6 %. Le Groupe témoigne d'un fort dynamisme en Europe, aux Etats-Unis et au Japon tandis que les pays d'Asie montrent des évolutions contrastées. Au quatrième trimestre, les ventes augmentent de 12 % par rapport à la même période de 2014. La croissance organique s'établit à 5 %. Le résultat opérationnel courant s'établit à 6 605 millions d'euros en 2015, en hausse de 16 %, une évolution à laquelle participe l'ensemble des métiers. Le résultat net part du Groupe s'élève pour sa part à 3 573 millions d'euros. Excluant la plus-value réalisée en 2014 suite à la distribution en nature des actions Hermès, le résultat net part du Groupe est en progression de 20 %.

Table A.2: LVMH 2015 gold standard summaries

Annual report	**(Carrefour 2013)**
Gold Standard #1	Communiqué de presse 5 mars 2014 RESULTATS ANNUELS 2013 : UNE DYNAMIQUE DE CROISSANCE Croissance confirmée des ventes en France et à l'international Progression du résultat opérationnel courant de +9,8% Multiplication par 6 du résultat net des activités poursuivies Dividende proposé en hausse, à 0,62€ par action Forte croissance des résultats du Groupe à changes constants • Chiffre d'affaires hors taxes de 74,9 Md€, en croissance de +2,5% hors essence • Hausse du résultat opérationnel courant : +9,8%, à 2 238 M€ • Multiplication par 6,3 du résultat net des activités poursuivies, part du Groupe : 949 M€ Europe : croissance du résultat opérationnel courant de +11,3% France : progression dans tous les formats • Retour à la croissance organique hors essence du chiffre d'affaires : +1,0% • Amélioration de l'image-prix, augmentation de la satisfaction des clients, hausse des débits • Forte hausse du résultat opérationnel courant : +30% Autres pays d'Europe : rebond au second semestre • Redressement très sensible des performances au second semestre en Europe, particulièrement en Espagne Pays émergents : croissance du résultat opérationnel courant de +8,5% • Progression remarquable des ventes organiques au Brésil et en Argentine, où Carrefour conforte sa position de leader dans l'alimentaire • Accélération de la croissance organique en Asie, où Carrefour poursuit son expansion • Reprise des investissements et renforcement de la structure financière • Investissements de 2,2 Md€, en hausse de +44% • Amélioration de 10% du ratio d'endettement à 1,1x • Baisse de la dette nette de 203 M€, à 4,1 Md€ Progression du dividende • Dividende proposé en hausse : 0,62€ par action, payable en numéraire ou en titres 1 A taux de changes constants.

Table A.3: Carrefour 2013 gold standard summaries

Annual report	**(Essilor 2012 Sem 1)**
Gold Standard #1	Commentant ces résultats, Hubert Sagnières, Président-Directeur Général du groupe, a déclaré : Essilor a récolté au premier semestre les fruits de sa stratégie de croissance fondée notamment sur l'innovation produit et la conquête des marchés à forte croissance. La forte progression des résultats du groupe démontre une nouvelle fois la solidité de son modèle de création de valeur sur un marché en croissance structurelle. Dans un contexte économique globalement moins porteur au second semestre, la détermination des équipes à mettre en œuvre notre plan stratégique partout dans le monde et le lancement de Varilux® S series, une nouvelle génération de verres progressifs, nous rendent confiants dans la réalisation de nos objectifs annuels .
Gold Standard #2	Perspectives Fort d'un bon premier semestre et dans un contexte économique toujours incertain, le groupe confirme ses objectifs annuels, à savoir une croissance de son chiffre d'affaires hors change comprise entre 12% et 15% (entre 6% et 9% pour la croissance en base homogène et la croissance par acquisitions organiques), ainsi qu'un maintien du taux de contribution de l'activité à un niveau élevé hors effets des acquisitions stratégiques. Informations pratiques Une réunion d'analystes se tiendra ce jour, 31 août, à 9h45, à Paris. La retransmission de la réunion sera accessible en direct et en différé par le lien suivant : http://hosting.3sens.com/Essilor/20120831-209C2A55/fr/ La présentation sera accessible par le lien suivant : `http://www.essilor.com/fr/Actionnaires/Pages/RapportsPresentations.aspx` Information Réglementée : Le rapport financier semestriel est disponible sur le site www.essilor.com, en cliquant sur le lien suivant : `http://www.essilor.com/fr/Actionnaires/Pages/RapportsPresentations.aspx`.

Table A.4: Essilor 2012 gold standard summaries

Sem report	**(Scor 2014)**
Gold Standard #1	Denis Kessler, Président-Directeur général de SCOR, déclare : Le groupe dégage des résultats de grande qualité en 2014. Il a poursuivi sa croissance et élargi son fonds de commerce. La profitabilité technique de SCOR Global P&C - qui délivre un ratio combiné de 91,4 % -, et celle de SCOR Global Life – qui enregistre une marge technique de 7,1 % -, sont très satisfaisantes. En dépit de la faiblesse des taux d'intérêt, SCOR Global Investments enregistre un rendement des actifs de 2,9 %. SCOR délivre une nouvelle fois un niveau élevé de rentabilité et atteint un niveau de solvabilité en ligne avec ses objectifs stratégiques. Ayant investi dans des nouveaux outils de souscription et de modélisation des risques, le groupe SCOR s'est préparé au nouveau régime prudentiel Solvabilité II qui entre en vigueur au 1er janvier 2016. Il a confiance dans sa capacité à relever les défis d'un environnement financier difficile, d'une situation concurrentielle aiguisée et d'un nouveau régime prudentiel exigeant. .
Gold Standard #2	Communiqué de presse 5 mars 2015 • Le résultat net s'élève à EUR 512 millions, soit une hausse de 40 % par rapport à 2013, sans prise en compte du gain d'acquisition exceptionnel d'un montant de EUR 183 millions lié à Generali US. Le taux de rendement des capitaux propres (ROE) atteint 9,9 %. • Les fonds propres atteignent EUR 5 729 millions au 31 décembre 2014 contre EUR 4 980 millions au 31 décembre 2013. Après versement de dividendes à hauteur de EUR 243 millions, l'actif net comptable par action augmente de 15 % à EUR 30,60 au 31 décembre 2014, contre EUR 26,64 au 31 décembre 2013. • Le ratio de solvabilité de SCOR, tel que défini par le modèle interne 20143, se maintient légèrement au-dessus de la fourchette optimale. • Le ratio d'endettement financier de SCOR s'élève à 23,1 % au 31 décembre 2014. Le placement de deux dettes subordonnées à durée indéterminée pour un montant de EUR 250 millions et de CHF 125 millions4 en 2014 témoigne de la capacité du Groupe à gérer de façon dynamique le passif de son bilan. • SCOR proposera à l'Assemblée générale d'augmenter le dividende à EUR 1,40 par action pour 2014, contre EUR 1,30 en 2013, soit un taux de distribution de 51 %. Le dividende proposé pour 2014 sera détaché le 5 mai 2015 à EUR 1,40 et mis en paiement le 7 mai 2015.
Gold Standard #3	SCOR enregistre un résultat net de EUR 512 millions et une augmentation de 15 % de ses fonds propres, et propose un dividende de EUR 1,40 (+ 8 %) En 2014, le Groupe a enregistré, trimestre après trimestre, des résultats de très bonne qualité, qui lui ont permis d'améliorer sa position concurrentielle. Grâce à la diversification de son fonds de commerce, à sa dimension globale et à de solides performances financières, SCOR délivre une nouvelle fois en 2014 un niveau élevé de rentabilité et atteint un niveau de solvabilité en ligne avec ses objectifs stratégiques. Les primes brutes émises atteignent EUR 11 316 millions, en hausse de 10,4 %1 par rapport à 2013 (10,8 % à taux de change constants). Cette croissance s'explique par la contribution de Generali US après son acquisition par le Groupe en octobre 2013, et par la croissance tant de SCOR Global Life en Asie et dans les branches Solutions financières & Longévité, que par celle de SCOR Global P&C: o Les primes brutes émises par SCOR Global P&C augmentent de 2,7 % à taux de change constants (+1,8 % à taux de change courants) à EUR 4 935 millions ; o Les primes brutes émises par SCOR Global Life s'élèvent à EUR 6 381 millions, en hausse de 5,5 %, à taux de change constants comme à taux de change courants, sur une base pro forma.

Table A.5: Scor 2014 gold standard summaries

Annual report	**(Faurecia 2012)**
Gold Standard #1	Communiqué de presse Nanterre, le 23 octobre 2012 Hausse de 7,9% du chiffre d'affaires consolidé au 3ème trimestre. Fort ralentissement de la production automobile en Europe attendu au 4ème trimestre Chiffre d'affaires du troisième trimestre Le chiffre d'affaires consolidé de Faurecia s'élève, au 3ème trimestre 2012, à 4 086 millions d'euros, en hausse de 7,9% sur le 3ème trimestre 2011 et stable à données comparables. Les ventes de produits s'élèvent à 3 217 millions d'euros, en hausse de 3,2% à données comparables. Elles traduisent : • Une forte croissance des ventes en Amérique du Nord, en hausse de 19% ; • Une croissance soutenue en Asie et en Amérique du Sud, en hausse de 10% ; • Un repli significatif de 4% des ventes en Europe ; • Une progression des ventes du groupe supérieure à la production automobile sur chacun des grands marchés mondiaux ; • Un meilleur équilibre géographique avec des ventes hors d'Europe passant à 48% du chiffre d'affaires au troisième trimestre 2012 contre 40% pour la même période de 2011.

Table A.6: Faurecia 2012 gold standard summaries

Sem report	**(Fnac Darty 2021)**
Gold Standard #1	2.2 FAITS MARQUANTS ET INFORMATIONS DU SEMESTRE Fnac Darty enregistre un chiffre d'affaires de 3 465M€ au 1er semestre 2021, en progression de +21,6% en données publiées et de +21,3% en données comparablesErreur ! Signet non défini. par rapport au 1er semestre 2020 portée à la fois par une hausse des volumes et du panier moyen. Cette solide croissance repose sur la poursuite de la croissance des ventes en ligne à +7,1%, malgré un effet de base de comparaison très élevé l'année dernière pendant le 1er confinement, et sur la solide dynamique enregistrée en magasins de +27,8%., malgré des conditions d'exploitation encore pénalisées par la poursuite des restrictions sanitaires sur le semestre. Ainsi, en France, après la fermeture des centres commerciaux et commerces de plus de 20 000 m^2 puis de plus de 10 000 m^2 et l'instauration d'un couvre-feu en début d'année, un troisième confinement national a été mis en place du 3 avril au 18 mai inclus, entraînant la fermeture de plusieurs magasins du Groupe. En Belgique, le Gouvernement a, quant à lui, imposé un confinement strict du 27 mars au 11 mai inclus, entraînant la fermeture des commerces non essentiels, sans impact majeur pour les enseignes du Groupe. En Suisse, les magasins ont été fermés pendant un mois et demi à compter du 18 janvier. Enfin, en Péninsule Ibérique, des confinements régionaux ont été imposés avec le maintien des limitations de trafic, des restrictions horaires dans certains magasins et des fermetures pour les magasins des centres commerciaux.

Table A.7: Fnac Darty 2021 gold standard summaries

Annual report	**(Technicolor 2013)**
Gold Standard #1	Frederic Rose, Directeur général de Technicolor, a déclaré : Ce trimestre a été marqué par une croissance robuste de notre chiffre d'affaires résultant de nos efforts permanents en matière d'exécution. Cette bonne performance a reflété le niveau soutenu de nos revenus de Licences, la forte croissance du segment Maison Connectée, ainsi que la hausse du chiffre d'affaires de nos activités cœur dans les Services Entertainment. Nous avons augmenté nos parts de marché dans nos différents métiers, tout en poursuivant nos efforts en matière d'innovation pour soutenir la croissance de nos activités et renforcer notre propriété intellectuelle. Nous sommes en bonne voie pour atteindre nos objectifs 2013. Frederic Rose, Directeur général, et Stéphane Rougeot, Directeur Financier et SEVP Stratégie, tiendront une conférence téléphonique le vendredi 26 avril à 16h00 (heure de Paris, GMT+1).
Gold Standard #2	Chiffre d'affaires du premier trimestre 2013 : Croissance robuste de 2,2% à périmètre et taux de change constants Paris (France), le 26 avril 2013 – Le Conseil d'Administration de Technicolor (Euronext Paris : TCH) s'est réuni hier afin d'examiner le chiffre d'affaires du premier trimestre 2013 du Groupe (non-audité). Principaux éléments du chiffre d'affaires du premier trimestre 2013 Le chiffre d'affaires des activités poursuivies du Groupe a atteint 775 millions d'euros au premier trimestre 2013, en hausse à périmètre constant1 de 2,1% à taux de change courants et de 2,2% à taux de change constants par rapport au premier trimestre 20122. Technologie : Nouveau trimestre de solide chiffre d'affaires des activités de Licences, reflétant les bonnes performances des différents programmes de la division. Services Entertainment : Légère hausse du chiffre d'affaires (activités en fin d'exploitation exclues), reflétant la croissance des activités de Services DVD et la robuste performance des activités de Services Créatifs Numériques. Maison Connectée : Forte performance, en ligne avec la dynamique du second semestre 2012.

Table A.8: Technicolor 2013 gold standard summaries

Annual report	**(ENGIE 2016)**
Gold Standard #1	A l'occasion de la présentation des résultats annuels, Isabelle Kocher, Directeur Général d'ENGIE, a déclaré : Nos résultats pour 2016 sont solides, en ligne avec la guidance. Nous sommes en avance sur notre plan de transformation à 3 ans. En un an, nous avons déjà signé plus de 50% des cessions prévues et identifié 75% des investissements. Nous nous recentrons et accélérons notre développement sur nos métiers cœur : la production d'électricité bas carbone, les réseaux, principalement gaziers, et les solutions intégrées pour nos clients. Ces métiers stratégiques que nous maîtrisons parfaitement sont au cœur de la révolution énergétique et présentent des potentiels de croissance importants. En parallèle, nous développons de nouveaux moteurs de croissance en nous appuyant sur l'innovation et le digital. Notre plan de performance "Lean 2018" progresse également plus vite que prévu, nous conduisant à en relever l'objectif de 20%. Tous ces leviers nous permettent de confirmer notre objectif pour 2018 : un Groupe agile, moins carboné et au profil dérisqué, pour devenir leader de la transition énergétique dans le monde.
Gold Standard #2	Communiqué de presse 2 mars 2017 Résultats 2016 en ligne avec la guidance En avance sur le plan de transformation Accélération de la croissance organique en 2017 Des résultats 2016 conformes à la guidance • Le Groupe atteint sa guidance avec un résultat net récurrent part du Groupe de 2,5 milliards d'euros, bénéficiant des effets de son programme de performance et du redémarrage des centrales nucléaires belges, mais encore impacté par les prix des commodités: • Le résultat net part du Groupe est de - 0,4 milliard d'euros, du fait principalement de dépréciations liées à des pertes de valeur compensées en partie par des éléments non récurrents positifs ; • La génération de cash reste très soutenue à 9,7 milliards d'euros permettant une poursuite de la réduction de la dette nette.

Table A.9: Engie 2016 gold standard summaries

Appendix B

Chapter 6 appendix

B.1 Hyperparameters

Table B.1 contains the hyperparameters that were used for the fine-tuning experiments.

Parameter	**Value**
model name or path	⟨model path⟩
learning rate	5e-5
max target length	⟨depends on model⟩
max source length	⟨depends on model⟩
train batch size	1
valid batch size	1
num train epochs	4
overwrite output_dir	true
save strategy	steps
save steps	3000
evaluation strategy	steps
eval steps	1000
warmup ratio	0.1

Table B.1: Hyperparameters for the financial narrative summarisation pipeline

B.2 Statistical and Hypothetical Testing in NLP

This appendix is a knowledge base for basic statistics and hypothetical testing information that may help understand the parts where we perform statistical significance tests. We will define several basic terms in statistical testing:
Non-Parametric Tests are used when the distribution of the sample is unknown or is not normal. Non-parametric tests do not assume anything about the test statistic

distribution.
Sampling distribution is a distribution of all of the possible values of a sample statistic for a given sample size selected from a population. In the NLP use case, the population is a list of the evaluation metrics on the test split set.
Significance level, alpha or α , is a value or threshold that researchers should set before performing a statistical test. It defines how much the sample evidence must contradict the null hypothesis to be able to reject the null hypothesis for the entire population. For example, a significance level of 0.1 signifies a 10% risk of deciding that an effect exists when it does not exist.
Hypothetical testing is a set of statistical techniques that allow us to answer some questions about a population using only a sample of data from the population in question (The sample size should be enough to be able to generalise the assumption).
Risks in Decision Making:
- Type I Error: Reject a true null hypothesis. A type I error is a "false alarm". The probability of a Type I Error is called the level of significance of the test α.
- Type II Error: Failure to reject a false null hypothesis. Type II error represents a "missed true hypothesis". The probability of a Type II Error is β.

Steps of the hypothetical testing technique used in a critical value:

1. State the null hypothesis, H_0 and the alternative hypothesis H_1.
2. Choose the sample size n and the significance level α (based on the application use case and how much we can allow a false discovery).
3. Choose the appropriate statistical test to use based on the sample distribution.
4. Determine the critical values or the frontier separating the rejection and non-rejection regions.

One-tail test Vs two-tail test:

- $H_0 : \mu \geq \mu_o$ vs $H_1 : \mu < \mu_o$ (one-tail test, lower-tail)
- $H_0 : \mu \leq \mu_o$ vs $H_1 : \mu > \mu_o$ (one-tail test, upper-tail)
- $H_0 : \mu = \mu_o$ vs $H_1 : \mu \neq \mu_o$ (two-tail test)

Rejection Rules: consider test statistic z, and significance value α.

- Lower-tail test: Reject H_0 if $z \leq z_\alpha$
- Upper-tail test: Reject H_0 if $z \geq z_\alpha$
- Two-tail test: Reject H_0 if $|z| \geq z_{\frac{\alpha}{2}}$

B.3 Adversarial Analysis

model	R-1 / R	R-1 / P	R-1 / F	R-2 / R	R-2 / P	R-2 / F	R-3 / R	R-3 / P	R-3 / F
T5LONG	0.551	0.421	0.463	0.421	0.293	0.332	0.389	0.260	0.297
T5LONG-DROPPED-10	0.464	0.434	0.435	0.299	0.284	0.280	0.240	0.232	0.225
T5LONG-DROPPED-8	0.457	0.436	0.433	0.288	0.282	0.274	0.225	0.226	0.215
T5LONG-PERMUTED-10	0.550	0.421	0.463	0.337	0.235	0.266	0.282	0.189	0.216
T5LONG-DROPPED-6	0.444	0.439	0.428	0.269	0.278	0.263	0.198	0.217	0.197
T5LONG-PERMUTED-8	0.550	0.421	0.463	0.331	0.230	0.261	0.272	0.182	0.208
T5LONG-PERMUTED-6	0.550	0.421	0.463	0.317	0.221	0.250	0.252	0.169	0.193
T5LONG-MASKED-10	0.479	0.377	0.409	0.305	0.223	0.248	0.244	0.169	0.191
T5LONG-MASKED-8	0.469	0.367	0.400	0.293	0.212	0.237	0.227	0.155	0.176
T5LONG-DROPPED-4	0.414	0.446	0.416	0.225	0.269	0.236	0.145	0.197	0.159
T5LONG-PERMUTED-4	0.550	0.421	0.463	0.289	0.200	0.227	0.212	0.142	0.162
T5LONG-MASKED-6	0.458	0.355	0.388	0.271	0.193	0.217	0.195	0.130	0.149
T5LONG-MASKED-4	0.435	0.332	0.365	0.223	0.156	0.177	0.135	0.088	0.102

Table B.2: Ranking of the original summary against the different corrupted summaries (R1, R2, R3). DROPPED stands for word dropping, PERMUTED stands for word permutation, MASKED stands for Bert Mask Filling

model	R-S1 / R	R-S1 / P	R-S1 / F	R-S2 / R	R-S2 / P	R-S2 / F
T5LONG	0.421	0.293	0.332	0.418	0.289	0.328
T5LONG-DROPPED-10	0.299	0.284	0.280	0.300	0.291	0.283
T5LONG-DROPPED-8	0.288	0.282	0.274	0.289	0.289	0.277
T5LONG-PERMUTED-10	0.337	0.235	0.266	0.377	0.262	0.296
T5LONG-DROPPED-6	0.269	0.278	0.263	0.269	0.288	0.267
T5LONG-PERMUTED-8	0.331	0.230	0.261	0.374	0.259	0.294
T5LONG-PERMUTED-6	0.317	0.221	0.250	0.368	0.255	0.289
T5LONG-MASKED-10	0.305	0.223	0.248	0.308	0.223	0.249
T5LONG-MASKED-8	0.293	0.212	0.237	0.295	0.211	0.237
T5LONG-DROPPED-4	0.225	0.269	0.236	0.226	0.287	0.242
T5LONG-PERMUTED-4	0.289	0.200	0.227	0.355	0.246	0.279
T5LONG-MASKED-6	0.271	0.193	0.217	0.273	0.192	0.217
T5LONG-MASKED-4	0.223	0.156	0.177	0.227	0.156	0.178

Table B.3: Ranking of the original summary against the different corrupted summaries (R-S1, R-S2).

model	R-L / R	R-L / P	R-L / F	R-SU4 / R	R-SU4 / P	R-SU4 / F
T5LONG	0.520	0.416	0.451	0.468	0.338	0.378
T5LONG-PERMUTED-10	0.499	0.403	0.436	0.453	0.328	0.367
T5LONG-PERMUTED-8	0.498	0.402	0.435	0.452	0.327	0.366
T5LONG-DROPPED-10	0.458	0.429	0.433	0.364	0.354	0.346
T5LONG-PERMUTED-6	0.495	0.400	0.432	0.450	0.325	0.364
T5LONG-DROPPED-8	0.452	0.431	0.431	0.353	0.356	0.342
T5LONG-PERMUTED-4	0.489	0.396	0.427	0.445	0.322	0.360
T5LONG-DROPPED-6	0.441	0.433	0.427	0.335	0.360	0.334
T5LONG-DROPPED-4	0.416	0.438	0.416	0.296	0.370	0.317
T5LONG-MASKED-10	0.475	0.349	0.393	0.375	0.281	0.310
T5LONG-MASKED-8	0.467	0.339	0.384	0.363	0.269	0.299
T5LONG-MASKED-6	0.457	0.326	0.372	0.345	0.252	0.281
T5LONG-MASKED-4	0.438	0.302	0.349	0.307	0.221	0.248

Table B.4: Ranking of the original summary against the different corrupted summaries (R-L, R-SU4).

Appendix C

Chapter 7 appendix

C.1 Hyperparameters Pretraining Text2text Model

Explanation of all the hyperparameters used in this training:

- module_import (str): importing the task on which we will pretrain the model. In this case, it is an unsupervised task.
- mixture_name (str): task on which we will pretrain the model. In this case, it is an unsupervised task. and was imported from an external file.
- tpu (str): name of the TPU machine on which we will pretrain the model.
- gcp_project (str): GCP project used.
- tpu_zone (str): tpu zone.
- model_dir (str): directory where the last checkpoint of the starting training model is stored.
- gin_file (str): a configuration file used to pass different hyperparameters such as the architecture and config file of T5, the different objectives that could be used for pretraining.
- train_steps (int): number of pretraining steps. In this case, we use 1.5 Million. Since the initial model was pretrained for 1 Million steps, we will continue the pretraining for additional 500 k steps.
- save_checkpoints_steps (int): save the checkpoints every n steps.
- batch_size (int): batch size.
- tpu_topology (str): 2x2 : means tpu version V3.8.
- input_sequence_length(int, defaults to 512): maximum input length for every batch.

- target_sequence_length(int, defaults to 512): maximum output length for every batch.

C.2 Example of Arabic Financial News Summaries

	Document	أظهرت بيانات من شركة هاليفاكس للرهن العقاري، اليوم الأربعاء، أن أسعار المنازل البريطانية زادت في الشهر الماضي بأسرع وتيرة سنوية منذ يونيو حزيران ٢٠١٦ في أحدث مؤشر على ارتفاع سوق الإسكان بعد رفع إجراءات العزل العام المرتبطة بجائحة كورونا.، وارتفعت أسعار المساكن ٣.٧ % مقارنة بسبتمبر أيلول من العام الماضي، لتسجل تسارعا من معدل نمو سنوي بلغ ٢.٥ % في أغسطس آب. ، وقالت راسل جالي العضو المنتدب في هاليفاكس السياق مهم عند المقارنة السنوية، لكن سبتمبر أيلول ٢٠١٩ شهد ضبابية سياسية أثرت على السوقْ.، وقالت هاليفاكس إن أسعار المنازل زادت ٦.١ % بين أغسطس آب وسبتمبر أيلول.، كما تشير مؤشرات أخرى في سوق الإسكان إلى زيادة النشاط في الشهور القليلة الماضية بفعل إطلاق الطلب المكوبت بعد رفع إجراءات العزل العام وبفعل خفض ضريبة مبيعات العقارات.، لكن المحللين ما زالوا يشكون في أن ارتفاع الأسعار سيستمر. وقال جالي آلا نزال نعتقد أنه من المتوقع حدوث ضغوط تراجع واضحة على أسعار المنازل في وقت ما خلال الشهور المقبلة نظرا لأن الحقائق المرتبطة بركود اقتصادي باتت ملموسة على نحو أقوىْ.
TITLE	Gold	أسعار المنازل في بريطانيا ترتفع بأسرع وتيرة منذ ٢٠١٦ House prices in the UK are rising at the fastest rate since 2016
	mT5 small	أسعار المنازل البريطانية ترتفع بأسرع وتيرة منذ يونيو ٢٠١٦ UK house prices are rising at the fastest pace since June 2016
	araT5 small	أسعار المنازل البريطانية ترتفع بأسرع وتيرة منذ ٢٠١٦ UK house prices are rising at the fastest rate since 2016
	bert2bert	اسعار المنازل البريطانية تسجل قفزة سنوية في ديسمبر British house prices recorded an annual jump in December
	mT5 base	أسعار المنازل البريطانية ترتفع بأسرع وتيرة سنوية منذ ٢٠١٦ UK house prices are rising at the fastest annual rate since 2016
	araT5 base	أسعار المنازل البريطانية ترتفع بأسرع وتيرة منذ ٢٠١٦ UK house prices are rising at the fastest rate since 2016
	finaraT5 base	أسعار المنازل البريطانية ترتفع بأسرع وتيرة منذ ٢٠١٦ UK house prices are rising at the fastest rate since 2016
	mT5 large	أسعار المنازل البريطانية ترتفع بأسرع وتيرة منذ يونيو ٢٠١٦ UK house prices are rising at the fastest pace since June 2016

Table C.1: Different generated abstractive headlines showing the output of the various systems we trained. This article belongs to the "Real Estate" category.

	Document	تسببت مبيعات قوية للمستثمرين المصريين والأجانب في أن تواصل البورصة المصرية تراجعها خلال تعاملات اليوم، حيث سيطر اللون الأحمر على شاشات التداول، بعدما فشلت المشتريات الهزيلة للمستثمرين العرب في وقف موجة الخسائر.، ولجأت إدارة البورصة إلى وقف عدد كبير من الأسهم لتجاوزها النسب المحددة للصعود والهبوط والمحددة بنسبة ٥ %.، وأنهت البورصة المصرية، تعاملات جلسة اليوم الأربعاء، على تراجع جماعي، وخسر رأس المال السوقي لأسهم الشركات المدرجة نحو ١٠ مليارات جنيه ليغلق عند مستوى ٣٣٨٫٧٢٤ مليار جنيه.، على صعيد المؤشرات، تراجع المؤشر الرئيسي ءيجي إكس ٣٠آ بنسبة ٩٣٫٠ % ليغلق عند مستوى ١١٤٨٠ نقطة، كما نزل مؤشر ءيجي إكس ٥٠آ بنسبة ٣٣٫٢ % ليغلق عند مستوى ١٩٨٩ نقطة.، وتراجع مؤشر ءيجي إكس ٣٠آ محدد الأوزان بنسبة ٧٤٫٠ % ليغلق عند مستوى ١٣٤٥٢ نقطة. كما انخفض مؤشر ءيجي إكس ٣٠آ للعائد الكلي بنسبة ٩٫٠ % ليغلق عند مستوى ٤٤٩٦ نقطة.، وامتدت الخسائر إلى مؤشر أسهم الشركات المتوسطة والصغيرة ءيجي إكس ٧٠آ متساوي الأوزان الذي تراجع بنسبة ٩٧٫٢ % ليغلق عند مستوى ٢١٢١ نقطة. كما انخفض المؤشر الأوسع نطاقاً ءيجي إكس ١٠٠آ متساوي الأوزان بنسبة ٣٦٫٢ % لينهي تعاملات اليوم عند مستوى ٣٠٨٩ نقطة
TITLE	Gold	بورصة مصر باللون الأحمر.. ١٠ مليارات جنيه تبخرت من السوق The Egyptian Stock Exchange is in red.. 10 billion pounds have evaporated from the market
	mT5 small	مبيعات قوية تدفع بورصة مصر إلى خسائر ٥ % Strong sales push the Egyptian Stock Exchange to losses of 5%
	araT5 small	بورصة مصر تواصل الهبوط.. ومؤشرها يتراجع ٥ % The Egyptian Stock Exchange continues to decline...and its index is down 5%
	bert2bert	مبيعات المصريين في بورصة مصر تفقد ٥ تريليونات جنيه Sales of Egyptians in the Egyptian Stock Exchange lost 5 trillion pounds
	mT5 base	مبيعات مبيعات قوية للمستثمرين المصريين والأجانب تواصل تراجعها Strong sales sales to Egyptian and foreign investors continue to decline
	araT5 base	مبيعات عربية وأجنبية تدفع بورصة مصر للتراجع Arab and foreign sales push the Egyptian Stock Exchange to decline
	finaraT5 base	مبيعات عربية تكبد بورصة مصر ١٠ مليارات جنيه في دقائق Arab sales cost the Egyptian Stock Exchange 10 billion pounds in minutes
	mT5 large	مبيعات عربية تهوي ببورصة مصر من مستويات قياسية Arab sales are falling on the Egyptian Stock Exchange from record levels

Table C.2: Different generated abstractive headlines showing the output of the various systems we trained. This article belongs to the "financial markets" category. We provide English translations to provide context for the general readers. Translation may not reflect the real meaning.

	Document	قال وزير الطاقة السعودي، خالد الفالح، إنه اجتمع مع رئيس الوزراء العراقي، حيدر العبادي، وناقشا دور العراق في الاتفاق العالمي لخفض إنتاج النفط وتحقيق الاستقرار في أسواق الخام.، وفي حسابه على تويتر، قال الفالح إنه والعبادي ناقشا أهمية استمرار التنسيق بين البلدين المنتجين للنفط، واستثمارات في قطاعي الطاقة والصناعة وتشجيع الصادرات في كل من البلدين.، واجتمع الجانبان أثناء مشاركتهما في المنتدى الاقتصادي العالمي في منتجع دافوس بسويسرا. والسعودية والعراق هما أكبر منتجان للنفط في منظمة البلدان المصدرة للبترول (أوبك).
TITLE	Gold	السعودية والعراق يبحثان اتفاق خفض إنتاج النفط Saudi Arabia and Iraq are discussing an agreement to reduce oil production
	mT5 small	الفالح: ناقشا دور العراق في الاتفاق العالمي لخفض إنتاج النفط Al-Falih: They discussed Iraq's role in the global agreement to cut oil production.
	araT5 small	الفالح: اجتماع مع العراق بشأن اتفاق خفض الإنتاج Al-Falih: A meeting with Iraq regarding the production cut agreement
	bert2bert	السعودية والعراق يدعمان الاتفاق العالمي لخفض انتاج النفط Saudi Arabia and Iraq support the global agreement to reduce oil production.
	mT5 base	الفالح والعبادي يبحثان دور العراق في اتفاق النفط Al-Falih and Al-Abadi discuss Iraq's role in the oil agreement
	araT5 base	السعودية والعراق يناقشان دور العراق في اتفاق النفط Saudi Arabia and Iraq discuss Iraq's role in the oil agreement
	finaraT5 base	الفالح: السعودية والعراق تبحثان دور العراق في اتفاق النفط Al-Falih: Saudi Arabia and Iraq are discussing Iraq's role in the oil agreement
	mT5 large	الفالح والعبادي يبحثان دور العراق في اتفاق النفط Al-Falih and Al-Abadi discuss Iraq's role in the oil agreement

Table C.3: Different generated abstractive headlines showing the output of the various systems we trained. This article belongs to the "Oil & Gas" category. We provide English translations to provide context for the general readers. The translation may not reflect the exact meaning. In this example, the model was able to detect the name of the minister who commented on the negotiation and also, the model used colon in the right place, which is used to emphasise in the Arabic language.

C.3 Human Evaluation Task

Estimate the cost of the task and time needed: Before sharing the task with our human evaluators, we experimented with a native Arabic speaker. This experiment led to an average time cost of two to three minutes, depending on the length of the financial news.
Explanation of the best worst scale method: Best-Worst Scaling (BWS) (Parvin et al., 2016) is a method that requires individuals to pick out their favoured and least favourite selections or characteristics from a set of options. This technique has become popular for exploring and constructing decisions in different fields. Louviere et al. (2015) provides an extensive outline of BWS, with theories and approaches for various types of scenarios. It contains case studies showing straightforward yet reliable ways to design, implement, use, and analyse the information acquired from decision activities in various contexts. Additionally, it illustrates the wide range of potential applications across many disciplines. BWS offers an option to traditional rating scales. Its convenience of use, interpretability, and established measuring properties make it attractive for those who wish to measure subjective data. We should mention that the BWS method does not give weights to tasks, which means system 1 beats system 2 would have the same importance as system 1 beats system 4. Furthermore, system A was selected better than system B and system B was selected better than system C, but this does not mean that system A will be selected better than system C.
Arabic native speaker qualification: This qualification is for evaluators who are native speakers of Arabic and have spoken the language fluently since childhood. Native speakers have a thorough understanding of the language and its nuances and can accurately evaluate written or spoken Arabic materials.
Instructions given to human evaluators:

Dear Evaluator,

Thank you for considering this task. We are conducting a study to evaluate the quality of different news summaries in Arabic. Your input is crucial to this research, and we appreciate your help. In this task, you will be presented with a news article and a set of two summaries, and you are asked to select the best summary for each set. You should base your judgments on accuracy (does the summary contain accurate facts?), informativeness (is important information captured?), fluency (is the summary written in well-formed Arabic?), typos and succinctness. This is motivated by our task being an abstractive summarisation task. We want to check if the model is generating readable content.

Informativeness: measures whether the summary covers the critical information from the input article.

Accuracy: measures whether the model is generating accurate information existing in the original text.

Fluency: focuses on if the generated summary is grammatically correct.

Typos: focuses on if the generated summary has some typos.

Succinctness: measures whether the summary is concise and does not describe too many details.

The length is not used as a criterion: sometimes shorter summaries could be better.

Disclaimer: there may be some technical terms or concepts in the summaries that you are unfamiliar with. Please do your best to evaluate the summaries based on the information provided.

Your input is completely anonymous and will be used for research purposes only.

Thank you again for your participation. We appreciate your help in improving the quality of summary generation.

Best regards.

Ethics application: following the research ethics of the university[1], we had to apply through the Research Ethics Application Management System (REAMS) system[2], and we had the ethical approval by Lancaster University FST Ethics Committee [3].

Screenshots:

[1] https://www.lancaster.ac.uk/research/research-services/research-integrity-ethics--governance/research-ethics/

[2] https://www.lancaster.ac.uk/research/research-services/research-integrity-ethics--governance/research-ethics/reams-web-guidance-/

[3] https://www.lancaster.ac.uk/sci-tech/research/ethics/

(a) Human Evaluation Task 1

(b) Human Evaluation Task 2

Figure C.1: Some Examples of how the task is formatted

(c) Human Evaluation Task 3

(d) Human Evaluation Task 4

Figure C.1: Some Examples of how the task is formatted

C.4 Correlation Measures

Correlation - Pearson r: the Pearson correlation coefficient is a commonly used assessment for the linear connection between two variables with a gaussian distribution and is simply referred to as the "correlation coefficient". This coefficient comes from a least-squares fit and can have three results: 1 implying a perfect positive correlation, -1 implying a negative correlation, or 0 showing no relation between the variables. The Pearson correlation is formulated as follows:

$$\rho = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y} \tag{C.1}$$

and the estimate is

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{C.2}$$

Correlation - Kendall: Kendall's tau is a non-parametric test which compares the degree of similarity to which two variables rank data which makes it viable for non-normal distributed data. Kendall tau can be computed for both continuous and ordinal data. To sum up, Kendall's tau is distinct from Spearman's rho due to its more severe punishment of non-sequential (in terms of the ranked variables) misalignments.

$$\tau = \frac{c-d}{c+d} \tag{C.3}$$

where c is the number of concordant pairs and d is the number of discordant pairs.

Correlation - Spearman rho: the Spearman correlation coefficient (rho) can be seen as a rank-based equivalent of Pearson's correlation coefficient and is suitable for variables that are not normally distributed and have a non-linear relationship. It is computed by taking the ranks of the variables (Si = rank(x) and Ri = rank(y)) and applying the Pearson formula to the ranked data. The formula is as follows:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{C.4}$$

Where d_i is the difference between the ranks of corresponding values in two variables.

Appendix D

Chapter 8 appendix

D.1 Hyperparameters Summarisation

Parameter	**Value**
model name	⟨model_name⟩
learning rate	5e-5
warmup ratio	0.1
max target length	⟨depends_on_model⟩
max source length	⟨depends_on_model⟩
train batch size	3
valid batch size	2
num train epochs	10
evaluation strategy	epoch
save_strategy	epoch
evaluation strategy	epoch
predict with generate	true
overwrite_output_dir	true
logging strategy	epoch

Table D.1: Hyperparameters for the French financial monitoring experiment

D.2 GPU Energy Monitoring

Figure D.1: GPU energy monitoring of French Pre-trained Seq2Seq model(validation split)

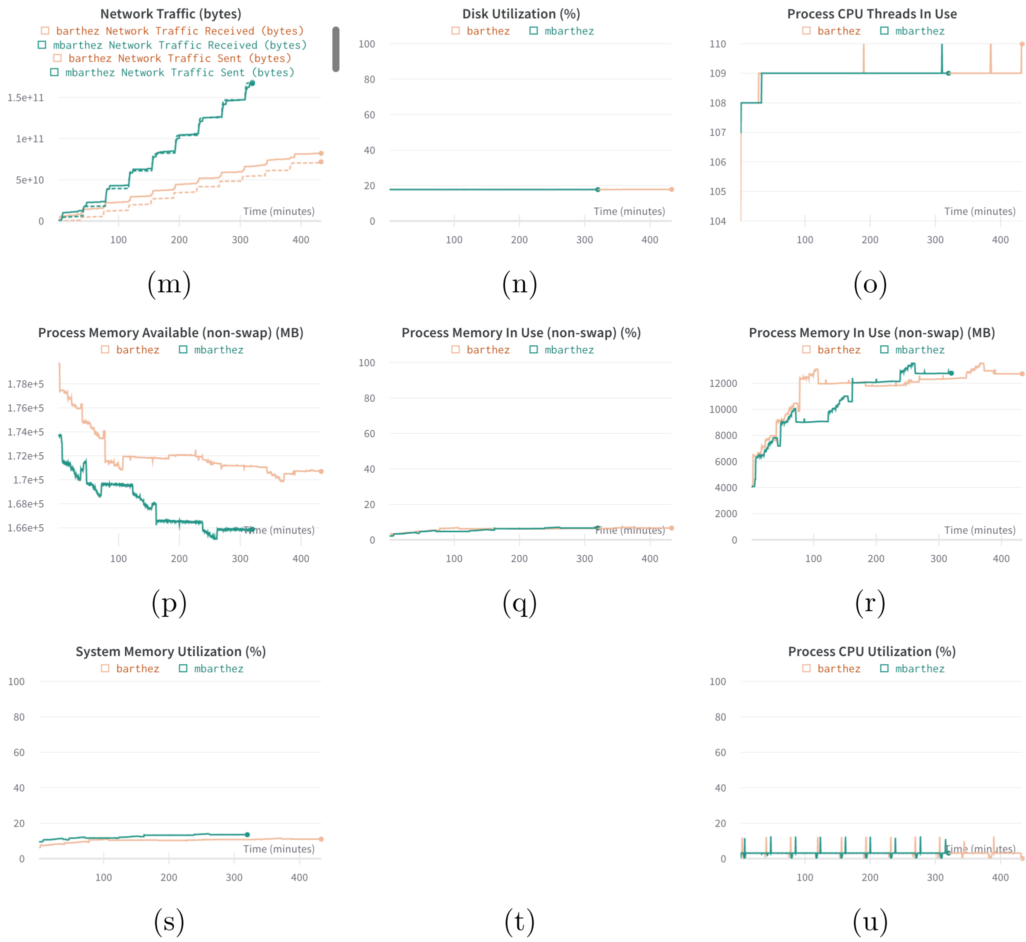


Figure D.1: GPU energy monitoring of French Pre-trained Seq2Seq model(validation split)

Publications

The following publications have been generated while developing this thesis:

Nadhem Zmandar, Mahmoud El-Haj, and Paul Rayson (2023a). "A Comparative Study of Evaluation Metrics for Long-Document Financial Narrative Summarization with Transformers". In: *Natural Language Processing and Information Systems*. Ed. by Elisabeth Métais, Farid Meziane, Vijayan Sugumaran, Warren Manning, and Stephan Reiff-Marganiec. Cham: Springer Nature Switzerland, pp. 391–403. ISBN: 978-3-031-35320-8

Nadhem Zmandar, Mahmoud El-Haj, and Paul Rayson (2021a). "Multilingual Financial Word Embeddings for Arabic, English and French". In: *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4584–4589

Nadhem Zmandar, Tobias Daudert, Sina Ahmadi, Mahmoud El-Haj, and Paul Rayson (June 2022). "CoFiF Plus: A French Financial Narrative Summarisation Corpus". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 1622–1639

Nadhem Zmandar, Mahmoud El-Haj, Paul Rayson, Ahmed Abura'Ed, Marina Litvak, Geroge Giannakopoulos, and Nikiforos Pittaras (2021e). "The Financial Narrative Summarisation Shared Task FNS 2021". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 120–125

Nadhem Zmandar, Abhishek Singh, Mahmoud El-Haj, and Paul Rayson (2021b). "Joint abstractive and extractive method for long financial document summarization". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 99–105

Nadhem Zmandar, Mo El-Haj, and Paul Rayson (2023b). "FinAraT5: A text to text model for financial Arabic text understanding and generation." In: *4th Conference on Language, Data and Knowledge*

Mahmoud El-Haj, **Nadhem Zmandar**, Paul Rayson, Ahmed AbuRa'ed, Marina Litvak, Nikiforos Pittaras, George Giannakopoulos, Aris Kosmopoulos, Blanca Carbajo-Coronado, and Antonio Moreno-Sandoval (June 2022d). "The Financial Narrative Summarisation Shared Task (FNS 2022)". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 43–52

Mahmoud El-Haj, Paul Rayson, and **Nadhem Zmandar**, eds. (June 2022b). *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association

Mahmoud El-Haj, Paul Rayson, and **Nadhem Zmandar**, eds. (2021a). *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics
Mahmoud El-Haj, Paul Rayson, Ismail El Maarouf, Najah-Imane Bentabet, Dominique Mariko, Estelle Labidurie, Marina Litvak, George Giannakopoulos, Ahmed AbuRa'ed, and **Nadhem Zmandar** (Dec. 2021b). "Review of the State of the Art in Financial Narrative Processing". English. In: *Financial Narrative Processing in Spanish*. Ed. by Antonio Moreno Sandoval. Tecnología, traducción y cultura. Tirant lo Blanch, pp. 51–98. ISBN: 9788418802423

Talks and Awards

Nadhem Zmandar was the presenter for all talks and presentations.

- Oral presentation at LREC conference (Marseille 2022).
- Oral presentation at Natural Language Processing & Information Systems (NLDB) 2023.
- Oral presentation at The Financial Narrative Processing Workshop (Lancaster 2021).
- Oral presentation at The Financial Narrative Processing Workshop (Marseille 2022).
- Oral presentation at International Conference on Big Data (Big Data).
- Nominated by the department for an FST Dean's Award 2nd year student prize.
- Participated in PhD speed talks at Lancaster University: Every PhD student has just 3 minutes to impress the audience with their research. This fast-paced session is always good, fun, and very informative.

Contribution Statements

(Zmandar et al., 2022): My work in this paper was to expand the Cofif corpus range from 2018 to 2021. Technically, I used a Python scraper to scrape the info-Financiere website and set custom rules to extract only companies from the cac40 or cac20. The other reports were presented and collected previously by the co-authors. Then I made a manual verification to delete the "document de reference" since they are not needed in this work. I only kept annual and bi-annual reports that had a clear and clean structure. Then I chose the gold standard candidates that will be used based on my financial knowledge. I wrote the heuristic rules to extract the gold standards

to ensure we cover most of the cases. The extraction script was based on some transformers and some other text, vocabulary and linguistic patterns. Then, I made a final manual check that took several weeks to verify the consistency of the summaries. The idea was to tweak every summary and verify that it was well-extracted. If not, I had to extract it manually myself to make sure I had a very good and clean corpus to be used later in my experiments. Finally, I prepared the statistics and plots describing the paper. I presented this paper in person at LREC 2022 in Marseille since the paper was selected for an oral presentation on stage.

(Zmandar et al., 2021b): My job consisted of extracting the sentences that maximise the Rouge metric in order to preprocess the dataset and perform the labelling, making the dataset suitable for training. In addition, I trained the ML and RL models on my private Google Colab Pro account. For the Reinforcement Learning pipeline, I defined a new reward function that maximises the Rouge-2 metrics to be aligned with the FNS task evaluation policy. Finally, I submitted the best version to the FNS 2021 challenge on the eval.ai evaluation platform. All the checkpoints, models, and summaries are available on the One Cloud storage of Lancaster University.

(El-Haj et al., 2022d): This paper presents the results and findings of the Financial Narrative Summarisation Shared Task 2022 on summarising UK, Greek and Spanish annual reports. This shared task is the third to target financial documents. The data for the shared task was created and collected from publicly available annual reports published by firms listed on the Stock Exchanges of UK, Greece and Spain. A total number of 14 systems from 7 different teams participated in the shared task. My contribution consisted in evaluating all the system-generated summaries in three languages and releasing the results. The team with the best ROUGE-2 scores for all three languages was selected as the winner of the competition. The scores are weighted as follows: English (50%), Spanish (25%) and Greek (25%).

(Zmandar et al., 2021e): The Financial Narrative Summarisation (FNS 2021 and FNS 2022) aimed to demonstrate the value and challenges of applying automatic text summarisation to financial text written in English, usually called financial narrative disclosures. For the financial summarisation shared task (FNS 2021 at FNP 2021), I configured the challenge on evala.ai[1] platform to automate the evaluation process of the task using a python celery worker, coded the evaluation script and configured the devops part of the python worker that will perform the queuing of submitted solutions and evaluate them against the gold summaries one by one using Rouge 2.0 package. I divided the challenge into two main parts: training and testing. I managed releasing results and presented the shared task during FNP 2021 (Lancaster UK). Using Eval-AI was very useful and enabled us to automate the evaluation of the submissions and to use custom evaluation phases and protocols. Eval-AI (Yadav et al., 2019) "is an open-source platform for evaluating and comparing Machine Learning (ML) and Artificial Intelligence (AI) algorithms. It is built to provide a scalable solution to the scientific research community and address the need to

[1] `https://eval.ai/web/challenges/challenge-page/1070/overview`

evaluate machine learning models by customisable metrics or through looping human evaluation. This helps researchers, students and data scientists to create, collaborate and participate in AI challenges organised around the world or by customising this platform and hosting it in a private cloud. This platform simplifies and standardises the process of bench-marking created models" (Yadav et al., 2019).

(El-Haj et al., 2021b): I participated in the writing of a book chapter: Review of the State of the Art in Financial Narrative Processing. The book was published by: Tirant lo Blanc (https://editorial.tirant.com/es/). The editor is: Prof Antonio Moreno Sandoval, UAM, Spain. My contribution was to write the three following subsections: *Financial Narratives*, *Types of Financial Narratives* and *Financial Narratives Structure*.

Financial Narrative Community Collaboration

During this thesis, we had the chance to collaborate with different members from the international financial NLP community.

- Dr George Giannakopoulos: NCSR Demokritos (FNS-Greek) : Chair of the MultiLing Summarisation workshop series.
- Prof Antonio Moreno – UAM. Madrid, Spain (FNS-Spanish).
- Abhishek Singh - Samsung, India (FNS-English).
- Tobias Daudert and Sina Ahmadi : Insight Centre for Data Analytics.

References

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Abdaljalil, Samir and Houda Bouamor (2021). "An Exploration of Automatic Text Summarization of Financial Reports". In: *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*. Online: -, pp. 1–7. URL: https://aclanthology.org/2021.finnlp-1.1 (visited on 01/10/2022).

Abdalla, Mohamed et al. (Feb. 2020). "Exploring the Privacy-Preserving Properties of Word Embeddings: Algorithmic Validation Study (Preprint)". In: URL: http://dx.doi.org/10.2196/preprints.18055.

Abdul-Mageed, Muhammad, AbdelRahim Elmadany, and El Moatez Billah Nagoudi (Aug. 2021). "ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 7088–7105. URL: https://aclanthology.org/2021.acl-long.551 (visited on 02/16/2022).

Abo, Mohamed Elhag Mohamed, Ram Gopal Raj, and Atika Qazi (2019). "A Review on Arabic Sentiment Analysis: State-of-the-Art, Taxonomy and Open Research Challenges". In: *IEEE Access* 7, pp. 162008–162024.

Abu El-khair, Ibrahim (Nov. 2016). *1.5 billion words Arabic Corpus*. Tech. rep. Publication Title: arXiv e-prints ADS Bibcode: 2016arXiv161104033A Type: article. URL: https://ui.adsabs.harvard.edu/abs/2016arXiv161104033A (visited on 03/08/2023).

Ait Azzi, Abderrahim and Juyeon Kang (Dec. 2020). "Extractive Summarization System for Annual Reports". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 143–147. URL: https://aclanthology.org/2020.fnp-1.24 (visited on 10/16/2021).

Al-Sabahi, Kamal, Zhang Zuping, and Yang Kang (2018a). *Bidirectional Attentional Encoder-Decoder Model and Bidirectional Beam Search for Abstractive Summarization*. arXiv: 1809.06662 [cs.CL].

Al-Sabahi, Kamal, Zhang Zuping, and Mohammed Nadher (2018b). "A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization

(HSSAS)". In: *IEEE Access* 6, pp. 24205–24212. URL: https://doi.org/10.1109%2Faccess.2018.2829199.

Alishahi, Afra, Marie Barking, and Grzegorz Chrupała (Aug. 2017). "Encoding of phonology in a recurrent neural model of grounded speech". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 368–378. URL: https://aclanthology.org/K17-1037.

Alishahi, Afra, Grzegorz Chrupała, and Tal Linzen (2019). *Analyzing and Interpreting Neural Networks for NLP: A Report on the First BlackboxNLP Workshop*. arXiv: 1904.04063 [cs.CL].

Alishahi, Afra et al., eds. (Nov. 2020). *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics. URL: https://aclanthology.org/2020.blackboxnlp-1.0.

Almazrouei, Ebtesam et al. (2023). *The Falcon Series of Open Language Models*. arXiv: 2311.16867 [cs.CL].

Alphonse, Pascal (2023). *XBRL, un nouveau standard de communication financière*. Working Papers. HAL. URL: https://EconPapers.repec.org/RePEc:hal:wpaper:hal-04138189.

Alwaqfi, Y. M. and M. S. Mohamad (2020). "A review of arabic optical character recognition techniques amp; performance". In: *International Journal of Engineering Trends and Technology*, pp. 44–51.

Alyafeai, Zaid and Maged Saeed (2020). *tkseem: A Preprocessing Library for Arabic.* https://github.com/ARBML/tnkeeh.

AMF France (2021). *Guide de l'information périodique des sociétés cotées.* URL: https://www.amf-france.org/sites/institutionnel/files/private/2021-09/2016-05-avril-2021.pdf.

Amini, M. R. et al. (2006). "Learning-based summarisation of xml documents". In: *Information Retrieval* 10 (3), pp. 233–255.

Anaconda Software Distribution (2020). Version Vers. 2-2.4.0. URL: https://docs.anaconda.com/.

Anil, Rohan et al. (2023). *PaLM 2 Technical Report*. arXiv: 2305.10403 [cs.CL]. URL: https://arxiv.org/abs/2305.10403.

Antoun, Wissam, Fady Baly, and Hazem Hajj (2020). "AraBERT: Transformer-based Model for Arabic Language Understanding". In: *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*. Marseille, France: European Language Resource Association, p. 9. URL: https://aclanthology.org/2020.osact-1.2.

— (Apr. 2021). "AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding". In: *Proceedings of the Sixth Arabic Natural Language Processing Workshop*. Kyiv, Ukraine (Virtual): Association for Computational Linguistics, pp. 191–195. URL: https://aclanthology.org/2021.wanlp-1.20 (visited on 03/08/2023).

Araci, Dogu (2019). "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models". In: *CoRR* abs/1908.10063. arXiv: 1908.10063. URL: http://arxiv.org/abs/1908.10063.

Arora, Piyush and Priya Radhakrishnan (Dec. 2020). "AMEX AI-Labs: An Investigative Study on Extractive Summarization of Financial Documents". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 137–142. URL: https://aclanthology.org/2020.fnp-1.23 (visited on 10/16/2021).

Ayed, Alaidine Ben, Ismaïl Biskri, and Jean-Guy Meunier (2021). "Arabic text summarization via Knapsack balancing of effective retention". In: *Procedia Computer Science* 189. AI in Computational Linguistics, pp. 312–319. ISSN: 1877-0509. URL: https://www.sciencedirect.com/science/article/pii/S1877050921012242.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv: 1409.0473 [cs.CL].

Bahja, Mohammed (2020). "Natural Language Processing Applications in Business". In: *E-Business*. Ed. by Robert M.X. Wu and Marinela Mircea. Rijeka: IntechOpen. Chap. 4. URL: https://doi.org/10.5772/intechopen.92203.

Baldeon Suarez, Jaime, Paloma Martínez, and Jose Luis Martínez (Dec. 2020). "Combining financial word embeddings and knowledge-based features for financial text summarization UC3M-MC System at FNS-2020". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 112–117. URL: https://aclanthology.org/2020.fnp-1.19 (visited on 10/16/2021).

Banerjee, Satanjeev and Alon Lavie (June 2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: https://aclanthology.org/W05-0909.

Barrios, Federico et al. (2016). "Variations of the Similarity Function of TextRank for Automated Summarization". In: *CoRR* abs/1602.03606. arXiv: 1602.03606. URL: http://arxiv.org/abs/1602.03606.

Bastings, Jasmijn et al., eds. (Nov. 2021). *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Punta Cana, Dominican Republic: Association for Computational Linguistics. URL: https://aclanthology.org/2021.blackboxnlp-1.0.

Bastings, Jasmijn et al., eds. (Dec. 2022). *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics. URL: https://aclanthology.org/2022.blackboxnlp-1.0.

Beech, D. G. (1962). “The Advanced Theory of Statistics. Volume 2, Inference and Relationship”. In: *Journal of the Royal Statistical Society Series C* 11.1, pp. 67–68. URL: https://ideas.repec.org/a/bla/jorssc/v11y1962i1p67-68.html.

Belkebir, Riadh and Ahmed Guessoum (2015). “A Supervised Approach to Arabic Text Summarization Using AdaBoost”. In: *New Contributions in Information Systems and Technologies*. Ed. by Alvaro Rocha et al. Cham: Springer International Publishing, pp. 227–236. ISBN: 978-3-319-16486-1.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan (Dec. 2020). “Longformer: The Long-Document Transformer”. In: *arXiv:2004.05150 [cs]*. arXiv: 2004.05150. URL: http://arxiv.org/abs/2004.05150 (visited on 03/01/2022).

Berger, Adam L. and Vibhu Mittal (2000). “OCELOT: a system for summarizing Web pages”. In: *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases*. Software available from wandb.com. URL: https://www.wandb.com/.

Bird, Steven, Ewan Klein, and Edward Loper (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* ” O’Reilly Media, Inc.”.

Bishara, Anthony J and James B. Hittner (2017). “Confidence intervals for correlations when data are not normal”. In: *Behavior Research Methods* 49, pp. 294–309.

Bishop, Jennifer, Qianqian Xie, and Sophia Ananiadou (May 2022). “GenCompareSum: a hybrid unsupervised summarization method using salience”. In: *Proceedings of the 21st Workshop on Biomedical Language Processing*. Dublin, Ireland: Association for Computational Linguistics, pp. 220–240. URL: https://aclanthology.org/2022.bionlp-1.22.

Bostrom, Kaj and Greg Durrett (2020). *Byte Pair Encoding is Suboptimal for Language Model Pretraining*. URL: https://arxiv.org/abs/2004.03720.

Brito, Eduardo et al. (2019). *Towards Supervised Extractive Text Summarization via RNN-based Sequence Classification*. arXiv: 1911.06121 [cs.CL].

Brown, Tom et al. (2020). “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cajueiro, Daniel O. et al. (2023). *A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding*. arXiv: 2301.03403 [cs.CL].

Cambria, Erik and Bebo White (2014). “Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]”. In: *IEEE Computational Intelligence Magazine* 9.2, pp. 48–57.

Cao, Yue et al. (2020). “MultiSumm: Towards a Unified Model for Multi-Lingual Abstractive Summarization”. In: *AAAI Conference on Artificial Intelligence*. URL: https://api.semanticscholar.org/CorpusID:213631816.

Carbonell, Jaime and Jade Goldstein (1998). “The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries”. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’98. Melbourne, Australia: Association for Computing Machinery, 335–336. ISBN: 1581130155. URL: https://doi.org/10.1145/290941.291025.

Celikyilmaz, Asli and Dilek Hakkani-Tur (July 2010). “A Hybrid Hierarchical Model for Multi-Document Summarization”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 815–824. URL: https://aclanthology.org/P10-1084.

CHALI, YLLIAS and SADID A. HASAN (2012). “Query-focused multi-document summarization: automatic data annotations and supervised learning approaches”. In: *Natural Language Engineering* 18.1, 109–145.

Chang, Chieh-Teng et al. (2018). *A Hybrid Word-Character Approach to Abstractive Summarization*. arXiv: 1802.09968 [cs.CL].

Chen, Pu-Chin et al. (Nov. 2021). “A Simple and Effective Positional Encoding for Transformers”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 2974–2988. URL: https://aclanthology.org/2021.emnlp-main.236.

Chen, Yen-Chun and Mohit Bansal (July 2018). “Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 675–686. URL: https://aclanthology.org/P18-1063.

Chen, Yifan et al. (Nov. 2020). “A Corpus of Very Short Scientific Summaries”. In: *Proceedings of the 24th Conference on Computational Natural Language Learning*. Ed. by Raquel Fernández and Tal Linzen. Online: Association for Computational Linguistics, pp. 153–164. URL: https://aclanthology.org/2020.conll-1.12.

Cheng, Jianpeng and Mirella Lapata (Aug. 2016). “Neural Summarization by Extracting Sentences and Words”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 484–494. URL: https://aclanthology.org/P16-1046.

Chiche, Alebachew and Betselot Yitagesu (Jan. 2022). “Part of speech tagging: a systematic review of deep learning and machine learning approaches”. In: *Journal of Big Data* 9.1. URL: https://doi.org/10.1186/s40537-022-00561-y.

Cho, Kyunghyun et al. (Oct. 2014). “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, pp. 103–111. URL: https://aclanthology.org/W14-4012.

Chopra, Sumit, Michael Auli, and Alexander M. Rush (June 2016). “Abstractive Sentence Summarization with Attentive Recurrent Neural Networks”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 93–98. URL: https://aclanthology.org/N16-1012.

Chouigui, Amina, Oussama Ben Khiroun, and Bilel Elayeb (2017). “ANT Corpus: An Arabic News Text Collection for Textual Classification”. In: *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 135–142.

Chowdhery, Aakanksha et al. (2022). *PaLM: Scaling Language Modeling with Pathways*. arXiv: 2204.02311 [cs.CL].

Chowdhury, G G (Dec. 2003). *Introduction to modern information retrieval*. en. 2nd ed. London, England: Facet Publishing.

Chowdhury, Shammur Absar et al. (Dec. 2020a). “Improving Arabic Text Categorization Using Transformer Training Diversification”. In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. Barcelona, Spain (Online): Association for Computational Linguistics, pp. 226–236. URL: https://aclanthology.org/2020.wanlp-1.21 (visited on 03/08/2023).

Chowdhury, Tanya, Sachin Kumar, and Tanmoy Chakraborty (2020b). *Neural Abstractive Summarization with Structural Attention*. arXiv: 2004.09739 [cs.CL].

Christopher, Olah. (2015). “Understanding LSTM Networks”. In: Accessed: 2021-10-19. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Chung, Hyung Won et al. (2022). *Scaling Instruction-Finetuned Language Models*. arXiv: 2210.11416 [cs.LG].

Cibils, André et al. (2018). *Diverse Beam Search for Increased Novelty in Abstractive Summarization*. arXiv: 1802.01457 [cs.CL].

Clark, Alexander, Chris Fox, and Shalom Lappin, eds. (June 2010). *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley. URL: https://doi.org/10.1002/9781444324044.

Clark, Kevin et al. (Mar. 2020). *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. arXiv:2003.10555 [cs]. URL: http://arxiv.org/abs/2003.10555 (visited on 03/08/2023).

Cohan, Arman et al. (June 2018). “A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 615–621. URL: https://aclanthology.org/N18-2097.

Collins, Ed, Isabelle Augenstein, and Sebastian Riedel (2017). *A Supervised Approach to Extractive Summarisation of Scientific Papers*. arXiv: 1706.03946 [cs.CL].

Colombo, Pierre, Chloe Clavel, and Pablo Piantanida (2021a). “InfoLM: A New Metric to Evaluate Summarization & Data2Text Generation”. In: *arXiv preprint arXiv:2112.01589*.

Colombo, Pierre et al. (2021b). “Automatic Text Evaluation through the Lens of Wasserstein Barycenters”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10450–10466.
Conneau, Alexis et al. (July 2020). “Unsupervised Cross-lingual Representation Learning at Scale”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747.
Conover, W and R Iman (Feb. 1979). *Multiple-comparisons procedures. Informal report*. URL: http://dx.doi.org/10.2172/6057803.
Cook, Shane (2012). *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 9780124159334.
Copara, Jenny et al. (June 2020). “Contextualized French Language Models for Biomedical Named Entity Recognition”. In: *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes*. Nancy, France: ATALA et AFCP, pp. 36–48. URL: https://aclanthology.org/2020.jeptalnrecital-deft.4.
Coppersmith, Glen and Erin Kelly (June 2014). “Dynamic Wordclouds and Vennclouds for Exploratory Data Analysis”. In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 22–29. URL: https://aclanthology.org/W14-3103.
Cui, Peng and Le Hu (June 2021). “Sliding Selector Network with Dynamic Memory for Extractive Summarization of Long Documents”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 5881–5891. URL: https://aclanthology.org/2021.naacl-main.470.
Dai, Z. and R. Huang (2021). “A joint model for structure-based news genre classification with application to text summarization”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.
Dai, Zihang et al. (2019). *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. arXiv: 1901.02860 [cs.LG].
Daudert, Tobias and Sina Ahmadi (Aug. 2019). “CoFiF: A Corpus of Financial Reports in French Language”. In: *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*. Macao, China, pp. 21–26. URL: https://aclanthology.org/W19-5504.
Devlin, Jacob et al. (June 2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis,

Minnesota: Association for Computational Linguistics, pp. 4171–4186. URL: https://aclanthology.org/N19-1423 (visited on 10/16/2021).
Dhouib, Amira et al. (2022). "Arabic Automatic Speech Recognition: A Systematic Literature Review". In: *Applied Sciences* 12.17. ISSN: 2076-3417. URL: https://www.mdpi.com/2076-3417/12/17/8898.
Di Fabbrizio, Giuseppe, Amanda Stent, and Robert Gaizauskas (June 2014). "A Hybrid Approach to Multi-document Summarization of Opinions in Reviews". In: *Proceedings of the 8th International Natural Language Generation Conference (INLG)*. Philadelphia, Pennsylvania, U.S.A.: Association for Computational Linguistics, pp. 54–63. URL: https://aclanthology.org/W14-4408.
Dogra, Varun et al. (June 2022). "A Complete Process of Text Classification System Using State-of-the-Art NLP Models". In: *Computational Intelligence and Neuroscience* 2022. Ed. by Sumarga Kumar Sah Tyagi, pp. 1–26. URL: https://doi.org/10.1155/2022/1883698.
Dohare, Shibhansh, Vivek Gupta, and Harish Karnick (July 2018). "Unsupervised Semantic Abstractive Summarization". In: *Proceedings of ACL 2018, Student Research Workshop*. Melbourne, Australia: Association for Computational Linguistics, pp. 74–83. URL: https://aclanthology.org/P18-3011.
Dohare, Shibhansh, Harish Karnick, and Vivek Gupta (2017). *Text Summarization using Abstract Meaning Representation*. arXiv: 1706.01678 [cs.CL].
Donahue, Chris, Mina Lee, and Percy Liang (July 2020). "Enabling Language Models to Fill in the Blanks". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 2492–2501. URL: https://aclanthology.org/2020.acl-main.225.
Dong, Yue et al. (2018). "BanditSum: Extractive Summarization as a Contextual Bandit". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3739–3748. URL: https://aclanthology.org/D18-1409.
Douka, Stella et al. (Nov. 2021). "JuriBERT: A Masked-Language Model Adaptation for French Legal Text". In: *Proceedings of the Natural Legal Language Processing Workshop 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 95–101. URL: https://aclanthology.org/2021.nllp-1.9 (visited on 01/04/2022).
Douzidia, Fouad Soufiane and Guy Lapalme (2004). "Lakhas, an Arabic summarization system". In: URL: https://api.semanticscholar.org/CorpusID:18307049.
Dror, Rotem et al. (July 2018). "The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1383–1392. URL: https://aclanthology.org/P18-1128.
Dunlavy, Daniel M. et al. (2007). "QCS: A system for querying, clustering and summarizing documents". In: *Information Processing Management* 43.6.

Text Summarization, pp. 1588–1605. ISSN: 0306-4573. URL: https://www.sciencedirect.com/science/article/pii/S0306457307000246.

Dunn, Olive Jean (1964). "Multiple Comparisons Using Rank Sums". In: *Technometrics* 6.3, pp. 241–252. eprint: https://www.tandfonline.com/doi/pdf/10.1080/00401706.1964.10490181. URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1964.10490181.

Durrett, Greg, Taylor Berg-Kirkpatrick, and Dan Klein (Aug. 2016). "Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1998–2008. URL: https://aclanthology.org/P16-1188.

Eddine, Moussa Kamal et al. (Oct. 2021). "FrugalScore: Learning Cheaper, Lighter and Faster Evaluation Metricsfor Automatic Text Generation". In: *arXiv:2110.08559 [cs]*. arXiv: 2110.08559. URL: http://arxiv.org/abs/2110.08559 (visited on 01/05/2022).

Eddine, Moussa Kamal et al. (Mar. 2022). "AraBART: a Pretrained Arabic Sequence-to-Sequence Model for Abstractive Summarization". In: *arXiv:2203.10945 [cs]*. arXiv: 2203.10945. URL: http://arxiv.org/abs/2203.10945 (visited on 04/21/2022).

Edmundson, H. P. (1969a). "New Methods in Automatic Extracting". In: *J. ACM* 16.2, 264–285. ISSN: 0004-5411. URL: https://doi.org/10.1145/321510.321519.

Edmundson, Harold P (1969b). "New methods in automatic extracting". In: *Journal of the ACM (JACM)* 16.2, pp. 264–285.

Efron, Bradley and Robert J. Tibshirani (1993). *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability 57. Boca Raton, Florida, USA: Chapman & Hall/CRC. URL: https://www.taylorfrancis.com/books/mono/10.1201/9780429246593/introduction-bootstrap-bradley-efron-tibshirani.

El-Haj, Dr Mahmoud et al., eds. (Dec. 2020a). *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING. URL: https://aclanthology.org/2020.fnp-1.0.

El-Haj, Mahmoud (2012). *Multi-document Arabic Text Summarisation.*

— (Sept. 2019). "MultiLing 2019: Financial Narrative Summarisation". In: *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*. Varna, Bulgaria: RANLP, pp. 6–10. URL: https://www.aclweb.org/anthology/W19-8902.

El-Haj, Mahmoud and Rim Koulali (2013). "KALIMAT a multipurpose Arabic corpus". In: URL: https://api.semanticscholar.org/CorpusID:62480828.

EL-Haj, Mahmoud, Udo Kruschwitz, and Chris Fox (July 2011). "Multi-Document Arabic Text Summarisation". In.

El-Haj, Mahmoud and Andrew Ogden (June 2022a). "Financial Narrative Summarisation Using a Hybrid TF-IDF and Clustering Summariser: AO-Lancs System at FNS 2022". In: *Proceedings of the 4th Financial Narrative Processing Workshop*

@LREC2022. Marseille, France: European Language Resources Association, pp. 79–82. URL: https://aclanthology.org/2022.fnp-1.11 (visited on 03/06/2023).

El-Haj, Mahmoud and Paul Rayson (May 2016). "OSMAN — A Novel Arabic Readability Metric". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 250–255. URL: https://aclanthology.org/L16-1038.

El-Haj, Mahmoud, Paul Rayson, and Andrew Moore (2018). "The First Financial Narrative Processing Workshop (FNP 2018)". In: *LREC 2018*.

El-Haj, Mahmoud, Paul Rayson, and Nadhem Zmandar, eds. (2021a). *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics.

— eds. (June 2022b). *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association.

El-Haj, Mahmoud et al. (May 2014). "Detecting Document Structure in a Very Large Corpus of UK Financial Reports". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 1335–1338. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/402_Paper.pdf.

El Haj, Mahmoud et al. (May 2018). "Towards a Multilingual Financial Narrative Processing System". English. In: *The First Financial Narrative Processing Workshop*. Ed. by Mahmoud El-Haj, Paul Rayson, and Andrew Moore. The 1st Financial Narrative Processing Workshop in LREC 2018, FNP 2018 ; Conference date: 07-05-2018, pp. 52–58. ISBN: 9791095546238. URL: http://wp.lancs.ac.uk/cfie/.

El-Haj, Mahmoud et al. (2019a). "In search of meaning: Lessons, resources and next steps for computational analysis of financial discourse". In: *Journal of Business Finance & Accounting* 46.3-4, pp. 265–306.

El-Haj, Mahmoud et al. (2019b). "Multilingual financial narrative processing: Analyzing annual reports in English, Spanish, and Portuguese". In: *Multilingual Text Analysis: Challenges, Models, And Approaches*. World Scientific, pp. 441–463.

El-Haj, Mahmoud et al., eds. (Sept. 2019c). *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*. Turku, Finland: Linköping University Electronic Press. URL: https://aclanthology.org/W19-6400.

El-Haj, Mahmoud et al. (2020b). "Retrieving, classifying and analysing narrative commentary in unstructured (glossy) annual reports published as PDF files". In: *Accounting and Business Research* 50.1, pp. 6–34.

El-Haj, Mahmoud et al. (2020c). "The Financial Narrative Summarisation Shared Task (FNS 2020)". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pp. 1–12.

El-Haj, Mahmoud et al. (Dec. 2020d). "The Financial Narrative Summarisation Shared Task (FNS 2020)". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain

(Online): COLING, pp. 1–12. URL: https://aclanthology.org/2020.fnp-1.1 (visited on 10/16/2021).
El-Haj, Mahmoud et al. (Dec. 2020e). "The Financial Narrative Summarisation Shared Task (FNS 2020)". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 1–12. URL: https://aclanthology.org/2020.fnp-1.1.
El-Haj, Mahmoud et al. (2020f). "The Financial Narrative Summarisation Shared Task (FNS 2020)". In: *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020)*. Barcelona, Spain.
El-Haj, Mahmoud et al. (Dec. 2021b). "Review of the State of the Art in Financial Narrative Processing". English. In: *Financial Narrative Processing in Spanish*. Ed. by Antonio Moreno Sandoval. Tecnología, traducción y cultura. Tirant lo Blanch, pp. 51–98. ISBN: 9788418802423.
El-Haj, Mahmoud et al. (2021c). "The Financial Narrative Summarisation Shared Task (FNS 2021)". In: *The Third Financial Narrative Processing Workshop (FNP 2021)*. Lancaster, UK.
El-Haj, Mahmoud et al. (June 2022c). "The Financial Narrative Summarisation Shared Task (FNS 2022)". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 43–52. URL: https://aclanthology.org/2022.fnp-1.6 (visited on 03/06/2023).
El-Haj, Mahmoud et al. (June 2022d). "The Financial Narrative Summarisation Shared Task (FNS 2022)". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 43–52.
El-Khair, Ibrahim Abu (2016). "1.5 billion words Arabic Corpus". In: *CoRR* abs/1611.04033. arXiv: 1611.04033. URL: http://arxiv.org/abs/1611.04033.
Elamir, Elsayed (2022). *On Uses of Van der Waerden Test: A Graphical Approach*. arXiv: 2203.02148 [stat.ME].
Elsaid, asmaa et al. (Mar. 2023). "A Hybrid Arabic text summarization Approach based on Seq-to-seq and Transformer". In: URL: https://doi.org/10.21203/rs.3.rs-2672691/v1.
El-Haj, Mahmoud et al. (Mar. 2019). "In search of meaning: Lessons, resources and next steps for computational analysis of financial discourse". In: *Journal of Business Finance amp; Accounting* 46.3–4, 265–306. ISSN: 1468-5957. URL: http://dx.doi.org/10.1111/jbfa.12378.
Erkan, Günes and Dragomir R. Radev (2004a). "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization". In: *J. Artif. Intell. Res.* 22, pp. 457–479.
Erkan, Günes and Dragomir R Radev (2004b). "Lexrank: Graph-based lexical centrality as salience in text summarization". In: *Journal of artificial intelligence research* 22, pp. 457–479.
Fan, Angela, David Grangier, and Michael Auli (July 2018). "Controllable Abstractive Summarization". In: *Proceedings of the 2nd Workshop on Neural Machine*

Translation and Generation. Melbourne, Australia: Association for Computational Linguistics, pp. 45–54. URL: https://aclanthology.org/W18-2706.

Fan, Y. et al. (2021). "Discovering representation sprachbund for multilingual pre-training". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Farghaly, Ali and Khaled Shaalan (2009). "Arabic Natural Language Processing: Challenges and Solutions". In: *ACM Transactions on Asian Language Information Processing* 8.4. ISSN: 1530-0226. URL: https://doi.org/10.1145/1644879.1644881.

Fatima, Noureen et al. (2022). "A Systematic Literature Review on Text Generation Using Deep Neural Network Models". In: *IEEE Access* 10, pp. 53490–53503.

Filippova, Katja et al. (2009). "Company-oriented extractive summarization of financial news". en. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09*. Athens, Greece: Association for Computational Linguistics, pp. 246–254. URL: http://portal.acm.org/citation.cfm?doid=1609067.1609094 (visited on 10/08/2021).

Filippova, Katja et al. (Sept. 2015). "Sentence Compression by Deletion with LSTMs". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 360–368. URL: https://aclanthology.org/D15-1042.

Firth, J. R. (1957). "Applications of General Linguistics". In: *Transactions of the Philological Society* 56.1, pp. 1–14. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-968X.1957.tb00568.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-968X.1957.tb00568.x.

Fiszman, Marcelo et al. (2009). "Automatic summarization of MEDLINE citations for evidence-based medical treatment: A topic-oriented evaluation". In: *Journal of Biomedical Informatics* 42.5. Biomedical Natural Language Processing, pp. 801–813. ISSN: 1532-0464. URL: https://www.sciencedirect.com/science/article/pii/S1532046408001263.

Flamary, Rémi et al. (2021). "POT: Python Optimal Transport". In: *Journal of Machine Learning Research* 22.78, pp. 1–8. URL: http://jmlr.org/papers/v22/20-451.html.

Foroutan, Negar et al. (June 2022). "Multilingual Text Summarization on Financial Documents". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 53–58. URL: https://aclanthology.org/2022.fnp-1.7 (visited on 03/06/2023).

"Front Matter" (1945). In: *Biometrika* 33.3. ISSN: 00063444. URL: http://www.jstor.org/stable/2332296 (visited on 04/27/2023).

Galanis, Dimitrios and Prodromos Malakasiotis (2008). "AUEB at TAC 2008". In: *Theory and Applications of Categories*.

Gallina, Y., F. Boudin, and B. Daille (2020). "Large-scale evaluation of keyphrase extraction models". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*.

Ganesan, Kavita (2018). *ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks*. arXiv: 1803.01937 [cs.IR].

Gao, Jun et al. (2023). *Mask-then-Fill: A Flexible and Effective Data Augmentation Framework for Event Extraction*. arXiv: 2301.02427 [cs.CL].

Ge, Y. et al. (2021). "Baco: a background knowledge- and content-based framework for citing sentence generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Confer.*

Gehring, Jonas et al. (2017). *Convolutional Sequence to Sequence Learning*. arXiv: 1705.03122 [cs.CL].

Ghaddar, Abbas et al. (2021). "JABER: Junior Arabic BERt". In: *CoRR* abs/2112.04329. arXiv: 2112.04329. URL: https://arxiv.org/abs/2112.04329.

Ghallab, Abdullatif, Abdulqader Mohsen, and Yousef Ali (Jan. 2020). "Arabic Sentiment Analysis: A Systematic Literature Review". In: *Applied Computational Intelligence and Soft Computing* 2020, pp. 1–21. URL: https://doi.org/10.1155/2020/7403128.

Ghosh, Siddhartha (2009). "Application of natural language processing (NLP) Techniques in E–governance". In: *E-Government Development and Diffusion*. IGI Global, pp. 122–132.

Gillick, Dan and Benoit Favre (June 2009). "A Scalable Global Model for Summarization". In: *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*. Boulder, Colorado: Association for Computational Linguistics, pp. 10–18. URL: https://aclanthology.org/W09-1802.

GNU, P (2007). *Free Software Foundation. Bash (3.2. 48)[Unix shell program].*

Gokhan, Tuba, Phillip Smith, and Mark Lee (2021). "Extractive Financial Narrative Summarisation using SentenceBERT Based Clustering". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 94–98. URL: https://aclanthology.org/2021.fnp-1.18 (visited on 03/06/2023).

— (Oct. 2022). "GUSUM: Graph-based Unsupervised Summarization Using Sentence Features Scoring and Sentence-BERT". In: *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*. Gyeongju, Republic of Korea: Association for Computational Linguistics, pp. 44–53. URL: https://aclanthology.org/2022.textgraphs-1.5.

Goldberg, Yoav (2017). *Neural Network Methods for Natural Language Processing*. Springer International Publishing. URL: https://doi.org/10.1007/978-3-031-02165-7.

Goldberg, Yoav and Omer Levy (2014). "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method". In: *CoRR* abs/1402.3722. arXiv: 1402.3722. URL: http://arxiv.org/abs/1402.3722.

Gong, Yihong and Xin Liu (2001). "Generic text summarization using relevance measure and latent semantic analysis". In: *SIGIR '01*.

Grashchenkov, Kirill, Andrey Grabovoy, and Ildar Khabutdinov (2022). “A Method of Multilingual Summarization For Scientific Documents”. In: *2022 Ivannikov Ispras Open Conference (ISPRAS)*, pp. 24–30.
Guellil, Imane et al. (2021). “Arabic natural language processing: An overview”. In: *Journal of King Saud University - Computer and Information Sciences* 33.5, pp. 497–507. ISSN: 1319-1578. URL: https://www.sciencedirect.com/science/article/pii/S1319157818310553.
Gunel, Beliz et al. (2020). *Mind The Facts: Knowledge-Boosted Coherent Abstractive Text Summarization*. arXiv: 2006.15435 [cs.CL].
Gupta, Som and S. K. Gupta (2019). “Abstractive summarization: An overview of the state of the art”. In: *Expert Syst. Appl.* 121, pp. 49–65. URL: https://api.semanticscholar.org/CorpusID:57992117.
Haboush, Ahmad et al. (Jan. 2012). “Arabic Text Summerization Model Using Clustering Techniques”. In: *World Comput Sci Inf Technol J* 2.
Hardy, Hardy and Andreas Vlachos (2018). “Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 768–773. URL: https://aclanthology.org/D18-1086.
Hasan, Tahmid et al. (Aug. 2021). “XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 4693–4703. URL: https://aclanthology.org/2021.findings-acl.413.
Hasselqvist, Johan, Niklas Helmertz, and Mikael Kågebäck (2017). *Query-Based Abstractive Summarization Using Neural Networks*. arXiv: 1712.06100 [cs.CL].
Henighan, Tom et al. (2020). *Scaling Laws for Autoregressive Generative Modeling*. arXiv: 2010.14701 [cs.LG].
Hirao, Tsutomu et al. (2007). “Supervised automatic evaluation for summarization with voted regression model”. In: *Information Processing Management* 43.6. Text Summarization, pp. 1521–1535. ISSN: 0306-4573. URL: https://www.sciencedirect.com/science/article/pii/S0306457307000416.
Hoffmann, Jordan et al. (2022). *Training Compute-Optimal Large Language Models*. arXiv: 2203.15556 [cs.CL].
Honnibal, Matthew and Ines Montani (2017). “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing”. To appear.
Hovy, Eduard (June 2022). “972Text Summarization”. In: *The Oxford Handbook of Computational Linguistics*. Oxford University Press. ISBN: 9780199573691. eprint: https://academic.oup.com/book/0/chapter/358153681/chapter-pdf/45720086/oxfordhb-9780199573691-e-008.pdf. URL: https://doi.org/10.1093/oxfordhb/9780199573691.013.008.
Hovy, Eduard and Chin-Yew Lin (Oct. 1998). “Automated Text Summarization and the Summarist System”. In: *TIPSTER TEXT PROGRAM PHASE III:*

Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 197–214. URL: https://aclanthology.org/X98-1026.

Huang, Luyang, Lingfei Wu, and Lu Wang (2020). *Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward.* arXiv: 2005.01159 [cs.CL].

Hunter, John D. (2007). "Matplotlib: A 2D Graphics Environment". In: *Computing in Science Engineering* 9.3, pp. 90–95.

Inoue, Go et al. (Apr. 2021). "The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models". In: *Proceedings of the Sixth Arabic Natural Language Processing Workshop.* Kyiv, Ukraine (Virtual): Association for Computational Linguistics, pp. 92–104. URL: https://aclanthology.org/2021.wanlp-1.10 (visited on 03/08/2023).

Iqbal, Touseef and Shaima Qureshi (2022). "The survey: Text generation models in deep learning". In: *Journal of King Saud University - Computer and Information Sciences* 34.6, Part A, pp. 2515–2528. ISSN: 1319-1578. URL: https://www.sciencedirect.com/science/article/pii/S1319157820303360.

Jiang, Kai and Xi Lu (2020). "Natural Language Processing and Its Applications in Machine Translation: A Diachronic Review". In: *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, pp. 210–214.

Jiao, Xiaoqi et al. (Nov. 2020). "TinyBERT: Distilling BERT for Natural Language Understanding". In: *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, pp. 4163–4174. URL: https://aclanthology.org/2020.findings-emnlp.372.

Jing, Hongyan and Kathleen R. McKeown (1999). "The Decomposition of Human-Written Summary Sentences". In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR '99. Berkeley, California, USA: Association for Computing Machinery, 129–136. ISBN: 1581130961. URL: https://doi.org/10.1145/312624.312666.

Jurafsky, Dan and James H. Martin (2024). *Speech and Language Processing (3rd ed. draft).* URL: https://web.stanford.edu/~jurafsky/slp3/.

Jurafsky, Dan and James H. Martin (2009a). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition.* Upper Saddle River, N.J.: Pearson Prentice Hall. ISBN: 9780131873216 0131873210. URL: http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y.

Jurafsky, Daniel and James H. Martin (2009b). *Speech and Language Processing (2Nd Edition).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0131873210.

Kågebäck, Mikael et al. (Apr. 2014). "Extractive Summarization using Continuous Vector Space Models". In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC).* Gothenburg, Sweden: Association for Computational Linguistics, pp. 31–39. URL: https://aclanthology.org/W14-1504.

Kahla, Mram, Zijian Győző Yang, and Attila Novák (Sept. 2021). "Cross-lingual Fine-tuning for Abstractive Arabic Text Summarization". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. Held Online: INCOMA Ltd., pp. 655–663. URL: https://aclanthology.org/2021.ranlp-1.74.

Kalpić, Damir, Nikica Hlupić, and Miodrag Lovrić (2011). "Student's t-Tests". In: *International Encyclopedia of Statistical Science*. Ed. by Miodrag Lovric. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1559–1563. ISBN: 978-3-642-04898-2. URL: https://doi.org/10.1007/978-3-642-04898-2_641.

Kamal Eddine, Moussa, Antoine Tixier, and Michalis Vazirgiannis (Nov. 2021). "BARThez: a Skilled Pretrained French Sequence-to-Sequence Model". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 9369–9390. URL: https://aclanthology.org/2021.emnlp-main.740.

Kamal Eddine, Moussa et al. (May 2022). "FrugalScore: Learning Cheaper, Lighter and Faster Evaluation Metrics for Automatic Text Generation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 1305–1318. URL: https://aclanthology.org/2022.acl-long.93.

Kaplan, Jared et al. (2020). *Scaling Laws for Neural Language Models*. arXiv: 2001.08361 [cs.LG].

Kaushik, Darsh et al. (June 2021). "CNLP-NITS @ LongSumm 2021: TextRank Variant for Generating Long Summaries". In: *Proceedings of the Second Workshop on Scholarly Document Processing*. Online: Association for Computational Linguistics, pp. 103–109. URL: https://aclanthology.org/2021.sdp-1.13.

Kazemnejad, Amirhossein et al. (2023). *The Impact of Positional Encoding on Length Generalization in Transformers*. arXiv: 2305.19466 [cs.CL].

Kendall, M. G. (1938). "A New Measure of Rank Correlation". In: *Biometrika* 30.1/2, pp. 81–93. ISSN: 00063444. URL: http://www.jstor.org/stable/2332226.

Keneshloo, Yaser et al. (2019). *Deep Reinforcement Learning For Sequence to Sequence Models*. arXiv: 1805.09461 [cs.LG].

Kennedy, Alistair and Stan Szpakowicz (2010). "Toward a Gold Standard for Extractive Text Summarization". In: *Canadian Conference on AI*.

Khanna, Urvashi et al. (June 2022). "Transformer-based Models for Long Document Summarisation in Financial Domain". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 73–78. URL: https://aclanthology.org/2022.fnp-1.10 (visited on 03/06/2023).

Khatri, Chandra, Gyanit Singh, and Nish Parikh (2018). "Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks". In: *ArXiv* abs/1807.08000. URL: https://api.semanticscholar.org/CorpusID:49907670.

Khurana, Diksha et al. (2022). “Natural language processing: state of the art, current trends and challenges”. In: *Multimedia Tools and Applications* 82.3, pp. 3713–3744. URL: https://doi.org/10.1007%2Fs11042-022-13428-4.

Kim, Tae Kyun (2015). “T test as a parametric statistic”. In: *Korean Journal of Anesthesiology* 68.6, p. 540. URL: https://doi.org/10.4097/kjae.2015.68.6.540.

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. URL: https://arxiv.org/abs/1412.6980.

Kiritchenko, Svetlana and Saif Mohammad (July 2017). “Best-Worst Scaling More Reliable than Rating Scales: A Case Study on Sentiment Intensity Annotation”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 465–470. URL: https://aclanthology.org/P17-2074.

Kitaev, Nikita, Lukasz Kaiser, and Anselm Levskaya (2020). “Reformer: The Efficient Transformer”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=rkgNKkHtvB.

Kiyoumarsi, Farshad (2015). “Evaluation of Automatic Text Summarizations based on Human Summaries”. In: *Procedia - Social and Behavioral Sciences* 192. The Proceedings of 2nd Global Conference on Conference on Linguistics and Foreign Language Teaching, pp. 83–91. ISSN: 1877-0428. URL: https://www.sciencedirect.com/science/article/pii/S1877042815034849.

Knight, Kevin and Daniel Marcu (2000). “Statistics-Based Summarization - Step One: Sentence Compression”. In: *AAAI/IAAI*.

— (July 2002). “Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression”. In: *Artif. Intell.* 139.1, 91–107. ISSN: 0004-3702. URL: https://doi.org/10.1016/S0004-3702(02)00222-9.

Koehn, P. (2010). *Statistical Machine Translation*. Statistical Machine Translation. Cambridge University Press. ISBN: 9780521874151. URL: https://books.google.co.uk/books?id=4v_Cx1wIMLkC.

Koehn, Philipp (July 2004). “Statistical Significance Tests for Machine Translation Evaluation”. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, pp. 388–395. URL: https://aclanthology.org/W04-3250 (visited on 02/03/2022).

Kokoska, Stephen and Daniel Zwillinger (1999). *CRC Standard Probability and Statistics Tables and Formulae, Student Edition*. URL: https://doi.org/10.1201/b16923.

Kouzis-Loukas, Dimitrios (2016). *Learning Scrapy*. Packt Publishing Ltd.

Kowalski, Charles J. (1972). “On the Effects of Non-Normality on the Distribution of the Sample Product-Moment Correlation Coefficient”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 21.1, pp. 1–12. ISSN:

00359254, 14679876. URL: http://www.jstor.org/stable/2346598 (visited on 04/25/2023).
Krimberg, Sophie, Natalia Vanetik, and Marina Litvak (2021). "Summarization of financial documents with TF-IDF weighting of multi-word terms". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 75–80. URL: https://aclanthology.org/2021.fnp-1.14 (visited on 03/06/2023).
Kuchmiichuk, P. (2023). "Silver data for coreference resolution in ukrainian: translation, alignment, and projection". In: *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*.
Kudo, Taku and John Richardson (Nov. 2018). "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, pp. 66–71. URL: https://aclanthology.org/D18-2012.
Kumawat, Deepika and Vinesh Jain (May 2015). "POS Tagging Approaches: A Comparison". In: *International Journal of Computer Applications* 118.6, pp. 32–38. URL: https://doi.org/10.5120/20752-3148.
Kupiec, Julian, Jan O. Pedersen, and Francine R. Chen (1995). "A trainable document summarizer". In: *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. URL: https://api.semanticscholar.org/CorpusID:5775833.
La Quatra, Moreno and Luca Cagliero (Dec. 2020). "End-to-end Training For Financial Report Summarization". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 118–123. URL: https://aclanthology.org/2020.fnp-1.20 (visited on 10/16/2021).
Laban, Philippe et al. (July 2020). "The Summary Loop: Learning to Write Abstractive Summaries Without Examples". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5135–5150. URL: https://aclanthology.org/2020.acl-main.460.
Lacoste, Alexandre et al. (2019). "Quantifying the Carbon Emissions of Machine Learning". In: *arXiv preprint arXiv:1910.09700*.
Ladhak, Faisal et al. (Nov. 2020). "WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4034–4048. URL: https://aclanthology.org/2020.findings-emnlp.360.
Lagrini, Samira and Mohammed Redjimi (2021). "A New Approach for Arabic Text Summarization". In: *Proceedings of the Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*. Trento, Italy:

Association for Computational Linguistics, pp. 176–185. URL: https://aclanthology.org/2021.icnlsp-1.20.
Lan, Zhenzhong et al. (Feb. 2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. arXiv:1909.11942 [cs]. URL: http://arxiv.org/abs/1909.11942 (visited on 03/08/2023).
Launay, Julien et al. (June 2022). "PAGnol: An Extra-Large French Generative Model". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4275–4284. URL: https://aclanthology.org/2022.lrec-1.455.
Le, Hang et al. (May 2020). "FlauBERT: Unsupervised Language Model Pre-training for French". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 2479–2490. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.302 (visited on 11/29/2021).
Lebanoff, Logan et al. (July 2020). "Understanding Points of Correspondence between Sentences for Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Online: Association for Computational Linguistics, pp. 191–198. URL: https://aclanthology.org/2020.acl-srw.26.
Lewis, Mike et al. (Oct. 2019). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *arXiv:1910.13461 [cs, stat]*. arXiv: 1910.13461. URL: http://arxiv.org/abs/1910.13461 (visited on 10/17/2021).
Lhoest, Quentin et al. (Nov. 2021). "Datasets: A Community Library for Natural Language Processing". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 175–184. arXiv: 2109.02846 [cs.CL]. URL: https://aclanthology.org/2021.emnlp-demo.21.
Li, Chenliang et al. (June 2018a). "Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 55–60. URL: https://aclanthology.org/N18-2009.
Li, Haoran et al. (Aug. 2018b). "Ensure the Correctness of the Summary: Incorporate Entailment Knowledge into Abstractive Sentence Summarization". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1430–1441. URL: https://aclanthology.org/C18-1121.
Li, Jing et al. (2022a). "A Survey on Deep Learning for Named Entity Recognition". In: *IEEE Transactions on Knowledge and Data Engineering* 34.1, pp. 50–70.
Li, Junyi et al. (2022b). *Pretrained Language Models for Text Generation: A Survey*. arXiv: 2201.05273 [cs.CL].

Li, Lei, Yafei Jiang, and Yinan Liu (Dec. 2020a). “Extractive Financial Narrative Summarisation based on DPPs”. In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 100–104. URL: https://aclanthology.org/2020.fnp-1.17 (visited on 10/16/2021).
Li, Piji, Lidong Bing, and Wai Lam (2018c). *Actor-Critic based Training Framework for Abstractive Summarization*. arXiv: 1803.11070 [cs.CL].
Li, Qian et al. (2021). *A Survey on Text Classification: From Shallow to Deep Learning*. arXiv: 2008.00364 [cs.CL].
Li, Wei et al. (2018d). “Improving Neural Abstractive Document Summarization with Explicit Information Selection Modeling”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1787–1796. URL: https://aclanthology.org/D18-1205.
Li, Wei et al. (July 2020b). “Leveraging Graph to Improve Abstractive Multi-Document Summarization”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6232–6243. URL: https://aclanthology.org/2020.acl-main.555.
Liao, Kexin, Logan Lebanoff, and Fei Liu (Aug. 2018). “Abstract Meaning Representation for Multi-Document Summarization”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1178–1190. URL: https://aclanthology.org/C18-1101.
Lierde, H. Van and Tommy W.S. Chow (2019). “Query-oriented text summarization based on hypergraph transversals”. In: *Information Processing &amp Management* 56.4, pp. 1317–1338. URL: https://doi.org/10.1016%2Fj.ipm.2019.03.003.
Lin, Chin-Yew (July 2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81. URL: https://aclanthology.org/W04-1013 (visited on 11/22/2021).
Lin, Jimmy, Nitin Madnani, and Bonnie Dorr (June 2010). “Putting the User in the Loop: Interactive Maximal Marginal Relevance for Query-Focused Summarization”. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 305–308. URL: https://aclanthology.org/N10-1041.
Lin, Zhouhan et al. (2017). *A Structured Self-attentive Sentence Embedding*. arXiv: 1703.03130 [cs.CL].
Linzen, Tal et al., eds. (Aug. 2019). *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics. URL: https://aclanthology.org/W19-4800.

Lippincott, Tom (Nov. 2018). “Portable, layer-wise task performance monitoring for NLP models”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 350–352. URL: https://aclanthology.org/W18-5445.

Litvak, Marina and Mark Last (Aug. 2013a). “Multilingual Single-Document Summarization with MUSE”. In: *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 77–81. URL: https://aclanthology.org/W13-3111.

Litvak, Marina and Natalia Vanetik (2013b). “Mining the gaps: Towards polynomial summarization”. In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 655–660.

— (2021). “Summarization of financial reports with AMUSE”. In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 31–36. URL: https://aclanthology.org/2021.fnp-1.5 (visited on 03/06/2023).

Liu, Fei et al. (2015). “Toward Abstractive Summarization Using Semantic Representations”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1077–1086. URL: https://aclanthology.org/N15-1114.

Liu, Feifan and Yang Liu (June 2008). “Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries”. In: *Proceedings of ACL-08: HLT, Short Papers*. Columbus, Ohio: Association for Computational Linguistics, pp. 201–204. URL: https://aclanthology.org/P08-2051.

Liu, Yang and Mirella Lapata (Nov. 2019a). “Text Summarization with Pretrained Encoders”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3730–3740. URL: https://aclanthology.org/D19-1387.

Liu, Yinhan et al. (July 2019b). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv:1907.11692 [cs]*. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692 (visited on 10/17/2021).

Lloret, Elena and Manuel Palomar (2011). “Text summarisation in progress: a literature review”. In: *Artificial Intelligence Review* 37, pp. 1 –41. URL: https://api.semanticscholar.org/CorpusID:254232944.

Lopyrev, Konstantin (2015). *Generating News Headlines with Recurrent Neural Networks*. arXiv: 1512.01712 [cs.CL].

Loughran, Tim and Bill Mcdonald (2011). “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks”. In: *The Journal of Finance* 66.1, pp. 35–65. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/29789771.

Louviere, Jordan J., Terry N. Flynn, and A. A. J. Marley (2015). *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.

Louviere, Jordan J. and George Woodworth (1983). "Design and Analysis of Simulated Consumer Choice or Allocation Experiments: An Approach Based on Aggregate Data". In: *Journal of Marketing Research* 20.4, pp. 350–367. ISSN: 00222437. URL: http://www.jstor.org/stable/3151440 (visited on 04/12/2023).

Luhn, H. P. (1958). "The Automatic Creation of Literature Abstracts". In: *IBM Journal of Research and Development* 2.2, pp. 159–165.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

Magooda, Ahmed and Cezary Marcjan (2020). *Attend to the beginning: A study on using bidirectional attention for extractive summarization*. arXiv: 2002.03405 [cs.CL].

Manaris, Bill (1998). "Natural Language Processing: A Human-Computer Interaction Perspective". In: *Advances in Computers*. Elsevier, pp. 1–66. URL: https://doi.org/10.1016/s0065-2458(08)60665-8.

Mani, Inderjeet et al. (June 1999). "The TIPSTER SUMMAC Text Summarization Evaluation". In: *Ninth Conference of the European Chapter of the Association for Computational Linguistics*. Bergen, Norway: Association for Computational Linguistics, pp. 77–85. URL: https://aclanthology.org/E99-1011.

Mani, Inderjeet et al. (2002). "SUMMAC: a text summarization evaluation". In: *Natural Language Engineering* 8, pp. 43 –68. URL: https://api.semanticscholar.org/CorpusID:17214063.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press. ISBN: 0262133601.

Martin, Louis et al. (July 2020). "CamemBERT: a Tasty French Language Model". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7203–7219. URL: https://aclanthology.org/2020.acl-main.645 (visited on 11/29/2021).

Masson, Corentin and Patrick Paroubek (May 2020). "NLP Analytics in Finance with DoRe: A French 250M Tokens Corpus of Corporate Annual Reports". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 2261–2267. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.275 (visited on 11/29/2021).

Mastropaolo, A. et al. (2021). "Studying the usage of text-to-text transfer transformer to support code-related tasks". In: URL: https://arxiv.org/abs/2102.02017.

Mckinney, Wes (2011). "pandas: a Foundational Python Library for Data Analysis and Statistics". In: URL: https://pandas.pydata.org/.

Mihalcea, Rada and Paul Tarau (2004). "Textrank: Bringing order into text". In: *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411.

Mikolov, Tomas et al. (2013). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119.
Miller, Derek (2019). *Leveraging BERT for Extractive Text Summarization on Lectures*. arXiv: 1906.04165 [cs.CL].
Mishra, Prabhaker et al. (2019). “Application of student's t-test, analysis of variance, and covariance”. In: *Annals of Cardiac Anaesthesia* 22.4, p. 407. URL: https://doi.org/10.4103/aca.aca_94_19.
Mohammed, Alshahrani et al. (2018). “BORSAH: An Arabic Sentiment Financial Tweets Corpus”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Mahmoud El-Haj, Paul Rayson, and Andrew Moore. Miyazaki, Japan: European Language Resources Association (ELRA). ISBN: 979-10-95546-23-8.
Mohit, Behrang (2014). “Named Entity Recognition”. In: *Natural Language Processing of Semitic Languages*. Ed. by Imed Zitouni. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 221–245. ISBN: 978-3-642-45358-8. URL: https://doi.org/10.1007/978-3-642-45358-8_7.
Mourri, Younes Bensouda, ukasz Kaiser, and Eddy Shyu (2021). *Natural Language Processing Specialization — deeplearning.ai*. https://www.deeplearning.ai/courses/natural-language-processing-specialization/.
Mridha, M. F. et al. (2021). “A Survey of Automatic Text Summarization: Progress, Process and Challenges”. In: *IEEE Access* 9, pp. 156043–156070.
Muniraj, Padhma et al. (2023). “HNTSumm: Hybrid text summarization of transliterated news articles”. In: *International Journal of Intelligent Networks* 4, pp. 53–61. ISSN: 2666-6030. URL: https://www.sciencedirect.com/science/article/pii/S2666603023000027.
Müller, Martin and Florian Laurent (2022). *Cedille: A large autoregressive French language model*. arXiv: 2202.03371 [cs.CL].
Nagoudi, El Moatez Billah, AbdelRahim Elmadany, and Muhammad Abdul-Mageed (May 2022a). “AraT5: Text-to-Text Transformers for Arabic Language Generation”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 628–647. URL: https://aclanthology.org/2022.acl-long.47 (visited on 03/20/2023).
Nagoudi, El Moatez Billah et al. (2022b). *JASMINE: Arabic GPT Models for Few-Shot Learning*. arXiv: 2212.10755 [cs.CL].
Najafi, M. and E. Tavan (2022). “Marsan at semeval-2022 task 6: isarcasm detection via t5 and sequence learners”. In: *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*.
Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou (2016a). *SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents*. arXiv: 1611.04230 [cs.CL].

Nallapati, Ramesh, Bowen Zhou, and Mingbo Ma (2016b). *Classify or Select: Neural Architectures for Extractive Document Summarization*. arXiv: 1611.04244 [cs.CL].
Nallapati, Ramesh et al. (Aug. 2016c). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pp. 280–290. URL: https://aclanthology.org/K16-1028.
Narayan, S., S. Cohen, and M. Lapata (2018a). "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization". In: URL: http://dx.doi.org/10.18653/v1/d18-1206.
Narayan, S., S. B. Cohen, and M. Lapata (2018b). "Ranking sentences for extractive summarization with reinforcement learning". In: *Proceedings of the 2018 Conference of the North American Chapter Of the Association for Computational Linguistics: Hu*.
Narayan, Shashi, Shay B. Cohen, and Mirella Lapata (2018c). "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1797–1807. URL: https://aclanthology.org/D18-1206.
Narayan, Shashi et al. (2017). *Neural Extractive Summarization with Side Information*. arXiv: 1704.04530 [cs.CL].
Narrain, Jigisha M et al. (2023). "Extractive Summarization - A Comparison of Pre-Trained Language Models and Proposing a Hybrid Approach". In: *2023 Winter Summit on Smart Computing and Networks (WiSSCoN)*, pp. 1–12.
Nema, Preksha et al. (July 2017). "Diversity driven attention model for query-based abstractive summarization". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1063–1072. URL: https://aclanthology.org/P17-1098.
Nenkova, Ani, Sameer Maskey, and Yang Liu (June 2011a). "Automatic Summarization". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Portland, Oregon: Association for Computational Linguistics, p. 3. URL: https://aclanthology.org/P11-5003.
Nenkova, Ani and Kathleen McKeown (2011b). "Automatic Summarization". In: *Foundations and Trends® in Information Retrieval* 5.2–3, pp. 103–233. ISSN: 1554-0669. URL: http://dx.doi.org/10.1561/1500000015.
— (2012). "A Survey of Text Summarization Techniques". In: *Mining Text Data*. URL: https://api.semanticscholar.org/CorpusID:556431.
Nikolov, Nikola I. and Richard Hahnloser (May 2020). "Abstractive Document Summarization without Parallel Data". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European

Language Resources Association, pp. 6638–6644. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.819.
Oliphant, T.E. (2015). *Guide to NumPy*. Continuum Press. ISBN: 9781517300074. URL: https://books.google.co.uk/books?id=g58ljgEACAAJ.
Orasan, Constantin (2019). "Automatic summarisation: 25 years On". In: *Natural Language Engineering* 25, pp. 735 –751. URL: https://api.semanticscholar.org/CorpusID:203708370.
Ortiz Suarez, Pedro and Simon Gabay (Oct. 2022). "A Data-driven Approach to Named Entity Recognition for Early Modern French". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 3722–3730. URL: https://aclanthology.org/2022.coling-1.327.
Ortiz Suárez, Pedro Javier et al. (May 2020). "Establishing a New State-of-the-Art for French Named Entity Recognition". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4631–4638. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.569.
Orzhenovskii, Mikhail (2021). "T5-LONG-EXTRACT at FNS-2021 Shared Task". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 67–69. URL: https://aclanthology.org/2021.fnp-1.12.
Ouchi, Hiroki, Hiroyuki Shindo, and Yuji Matsumoto (July 2017). "Neural Modeling of Multi-Predicate Interactions for Japanese Predicate Argument Structure Analysis". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1591–1600. URL: https://aclanthology.org/P17-1146.
Park, Jungyeul (May 2018). "Le benchmarking de la reconnaissance d'entités nommées pour le français (Benchmarking for French NER)". French. In: *Actes de la Conférence TALN. Volume 1 - Articles longs, articles courts de TALN*. Rennes, France: ATALA, pp. 241–250. URL: https://aclanthology.org/2018.jeptalnrecital-court.6.
Parker, Robert et al. (2009). "Arabic Gigaword Dataset." In: URL: https://catalog.ldc.upenn.edu/LDC2009T30.
Parnell, J., I. J. Unanue, and M. Piccardi (2021). "Rewardsofsum: exploring reinforcement learning rewards for summarisation". In: *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*.
Parvin, Shehely, Paul Wang, and Jashim Uddin (2016). "Using best-worst scaling method to examine consumers' value preferences: A multidimensional perspective". In: *Cogent Business & Management* 3.1. Ed. by Len Tiu Wright, p. 1199110. eprint: https://doi.org/10.1080/23311975.2016.1199110. URL: https://doi.org/10.1080/23311975.2016.1199110.
Pasunuru, Ramakanth and Mohit Bansal (June 2018). "Multi-Reward Reinforced Summarization with Saliency and Entailment". In: *Proceedings of the 2018*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, pp. 646–653. URL: https://aclanthology.org/N18-2102.

Paszke, Adam et al. (2017). "Automatic differentiation in PyTorch". In: *NIPS-W*.

Paul, Michael, ChengXiang Zhai, and Roxana Girju (Oct. 2010). "Summarizing Contrastive Viewpoints in Opinionated Text". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, pp. 66–76. URL: https://aclanthology.org/D10-1007.

Paulus, Romain, Caiming Xiong, and Richard Socher (2018). "A Deep Reinforced Model for Abstractive Summarization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HkAClQgA-.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct, pp. 2825–2830.

Penedo, Guilherme et al. (2023). *The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only*. arXiv: 2306.01116 [cs.CL].

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL: https://aclanthology.org/D14-1162.

Perez-Beltrachini, L. and M. Lapata (2021). "Models and datasets for cross-lingual summarisation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Pilault, Jonathan et al. (Nov. 2020). "On Extractive and Abstractive Neural Document Summarization with Transformer Language Models". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 9308–9319. URL: https://aclanthology.org/2020.emnlp-main.748.

Pomerleau, Dean A. (1990). "Neural Network Based Autonomous Navigation". In: *Vision and Navigation: The Carnegie Mellon Navlab*. Ed. by Charles E. Thorpe. Boston, MA: Springer US, pp. 83–93. ISBN: 978-1-4613-1533-9. URL: https://doi.org/10.1007/978-1-4613-1533-9_5.

PwC France (2019). *Financial Communication Framework And Practice*. URL: https://cliff.asso.fr/financial_communication_framework_and_practices/.

— (2020). *Financial Communication Framework And Practice*. URL: https://www.pwc.fr/fr/assets/files/pdf/2020/08/en-pwc-financial-communications-framework-and-practices-2020.pdf.

— (2021). *Financial Communication Framework And Practice*. URL: https://www.bredinprat.fr/wp-content/uploads/2021/07/COM-FI_GUIDE-2021_EN_pap_BD.pdf.

— (2022). *Financial Communication Framework And Practice.* URL: https://cliff.asso.fr/doc_site/source/OCF/2022_07_20_COM-FI_GUIDE_EN_FINAL.pdf.
Qiu, Liang, Yuanyi Ding, and Lei He (2018). *Recurrent Neural Networks with Pre-trained Language Model Embedding for Slot Filling Task.* arXiv: 1812.05199 [cs.CL].
R Core Team (2016). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.
Radford, Alec and Karthik Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training". In.
Rae, Jack W. et al. (2022). *Scaling Language Models: Methods, Analysis Insights from Training Gopher.* arXiv: 2112.11446 [cs.CL].
Raffel, Colin et al. (July 2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *arXiv:1910.10683 [cs, stat].* arXiv: 1910.10683. URL: http://arxiv.org/abs/1910.10683 (visited on 10/17/2021).
Raghu, Maithra et al. (2017). *On the Expressive Power of Deep Neural Networks.* arXiv: 1606.05336 [stat.ML].
Ramin, Kurt P and Cornelis Reiman (2013). *IFRS and XBRL: how to improve business reporting through technology and object tracking.* Vol. 1. Wiley Online Library.
Reda, Ameen et al. (2022). "A Hybrid Arabic Text Summarization Approach based on Transformers". In: *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 56–62.
Rehurek, Radim and Petr Sojka (2011). "Gensim–python framework for vector space modelling". In: *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3.2.
Reimers, Nils and Iryna Gurevych (Nov. 2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Hong Kong, China: Association for Computational Linguistics, pp. 3982–3992. URL: https://aclanthology.org/D19-1410.
— (Nov. 2020). "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.* Online: Association for Computational Linguistics. URL: https://aclanthology.org/2020.emnlp-main.365.
Richardson, Leonard (2007). "Beautiful soup documentation". In: URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.
Rosendahl, Jan et al. (2019). "Analysis of Positional Encodings for Neural Machine Translation". In: *Proceedings of the 16th International Conference on Spoken Language Translation.* Hong Kong: Association for Computational Linguistics. URL: https://aclanthology.org/2019.iwslt-1.20.
Rothe, Sascha, Shashi Narayan, and Aliaksei Severyn (2020). "Leveraging Pre-trained Checkpoints for Sequence Generation Tasks". In: *Transactions of the Association*

for Computational Linguistics 8, pp. 264–280. URL: https://aclanthology.org/2020.tacl-1.18.

Rush, Alexander M., Sumit Chopra, and Jason Weston (Sept. 2015). "A Neural Attention Model for Abstractive Sentence Summarization". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 379–389. URL: https://aclanthology.org/D15-1044.

Safaya, Ali (Aug. 2020). *Arabic-ALBERT*. URL: https://zenodo.org/record/4718724 (visited on 03/08/2023).

Safaya, Ali, Moutasem Abdullatif, and Deniz Yuret (Dec. 2020). "KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, pp. 2054–2059. URL: https://aclanthology.org/2020.semeval-1.271 (visited on 03/08/2023).

Saggion, H., K. Bontcheva, and H. Cunningham (2003). "Robust generic and query-based summarisation". In: *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - EACL '03*.

Saggion, Horacio and Thierry Poibeau (2012). "Automatic Text Summarization: Past, Present and Future". In: *Multi-source, Multilingual Information Extraction and Summarization*. Ed. by R. Yangarber T. Poibeau; H. Saggion. J. Piskorski. Theory and Applications of Natural Language Processing. Springer, pp. 3–13. URL: https://hal.science/hal-00782442.

Sai, Ananya B., Akash Kumar Mohankumar, and Mitesh M. Khapra (2022). "A Survey of Evaluation Metrics Used for NLG Systems". In: *ACM Comput. Surv.* 55.2. ISSN: 0360-0300. URL: https://doi.org/10.1145/3485766.

Saito, Itsumi et al. (2020). *Abstractive Summarization with Combination of Pre-trained Sequence-to-Sequence and Saliency Models*. arXiv: 2003.13028 [cs.CL].

Salton, Gerard and Christopher Buckley (1988). "Term-weighting approaches in automatic text retrieval". In: *Information Processing Management* 24.5, pp. 513–523. ISSN: 0306-4573. URL: https://www.sciencedirect.com/science/article/pii/0306457388900210.

Sanh, Victor et al. (2020). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv: 1910.01108 [cs.CL].

Schluter, Natalie (Apr. 2017). "The limits of automatic summarisation according to ROUGE". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 41–45. URL: https://aclanthology.org/E17-2007.

Schröder, Tim and Michael Schulz (2022). "Monitoring machine learning models: a categorization of challenges and methods". In: *Data Science and Management* 5.3, pp. 105–116. ISSN: 2666-7649. URL: https://www.sciencedirect.com/science/article/pii/S2666764922000303.

Schüller, Leon et al. (2020). *Windowing Models for Abstractive Summarization of Long Texts*. arXiv: 2004.03324 [cs.CL].

See, Abigail, Peter J. Liu, and Christopher D. Manning (July 2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1073–1083. URL: https://aclanthology.org/P17-1099.

Sellam, Thibault, Dipanjan Das, and Ankur Parikh (July 2020). "BLEURT: Learning Robust Metrics for Text Generation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7881–7892. URL: https://aclanthology.org/2020.acl-main.704 (visited on 02/17/2022).

Shaalan, Khaled (2010). In: *Machine Translation* 24.3/4, pp. 285–289. ISSN: 09226567, 15730573. URL: http://www.jstor.org/stable/41410952 (visited on 08/22/2023).

Shaalan, Khaled et al. (Sept. 2018). "Challenges in Arabic Natural Language Processing". In: *Computational Linguistics, Speech and Image Processing for Arabic Language*. WORLD SCIENTIFIC, pp. 59–83. URL: https://doi.org/10.1142/9789813229396_0003.

Shakhovska, Nataliya and Taras Cherna (2019). *The method of automatic summarization from different sources*. arXiv: 1905.02623 [cs.CL].

Shi, Tian et al. (Sept. 2020). "Neural Abstractive Text Summarization with Sequence-to-Sequence Models". en. In: *arXiv:1812.02303 [cs, stat]*. arXiv: 1812.02303. URL: http://arxiv.org/abs/1812.02303 (visited on 10/08/2021).

Shukla, Neelesh et al. (June 2022). "DiMSum: Distributed and Multilingual Summarization of Financial Narratives". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 65–72. URL: https://aclanthology.org/2022.fnp-1.9 (visited on 03/06/2023).

Simoulin, Antoine and Benoit Crabbé (June 2021). "Un modèle Transformer Génératif Pré-entrainé pour le______ français (Generative Pre-trained Transformer in______ (French) We introduce a French adaptation from the well-known GPT model)". French. In: *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*. Lille, France: ATALA, pp. 246–255. URL: https://aclanthology.org/2021.jeptalnrecital-taln.24.

Singh, Abhishek (Dec. 2020). "PoinT-5: Pointer Network and T-5 based Financial Narrative Summarisation". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 105–111. URL: https://aclanthology.org/2020.fnp-1.18 (visited on 10/16/2021).

Singh, Abhishek Kumar, Manish Gupta, and Vasudeva Varma (2017). "Hybrid MemNet for Extractive Summarization". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM. URL: https://doi.org/10.1145%2F3132847.3133127.

Sinha, Aakash, Abhishek Yadav, and Akshay Gahlot (2018). *Extractive Text Summarization using Neural Networks*. arXiv: 1802.10137 [cs.CL].

Smith, Shaden et al. (2022). *Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model*. arXiv: 2201.11990 [cs.CL].

Søgaard, Anders et al. (June 2014). "What's in a p-value in NLP?" In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 1–10. URL: https://aclanthology.org/W14-1601.

Song, Kaiqiang, Lin Zhao, and Fei Liu (Aug. 2018). "Structure-Infused Copy Mechanisms for Abstractive Summarization". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1717–1729. URL: https://aclanthology.org/C18-1146.

Song, Kaitao et al. (2019). *MASS: Masked Sequence to Sequence Pre-training for Language Generation*. arXiv: 1905.02450 [cs.CL].

Sornlertlamvanich, Virach (Dec. 2014). "Social Media Understanding by Word Cloud Timeline". In: *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*. Phuket,Thailand: Department of Linguistics, Chulalongkorn University, pp. 2–4. URL: https://aclanthology.org/Y14-1002.

Srinivasan, A. et al. (2021). "Predicting the performance of multilingual nlp models". In: eprint: 2110.08875. URL: https://arxiv.org/abs/2110.08875.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Staerman, Guillaume et al. (2021). "A pseudo-metric between probability distributions based on depth-trimmed regions". In: *arXiv preprint arXiv:2103.12711*.

Steinberger, Josef and Karel Ježek (2005). "Text Summarization and Singular Value Decomposition". In: *Advances in Information Systems*. Ed. by Tatyana Yakhno. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 245–254. ISBN: 978-3-540-30198-1.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum (July 2019). "Energy and Policy Considerations for Deep Learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3645–3650. URL: https://aclanthology.org/P19-1355.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014a). "Sequence to Sequence Learning with Neural Networks". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 3104–3112.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014b). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc. URL:

https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Suárez, Pedro Javier Ortiz, Benoît Sagot, and Laurent Romary (July 2019). "Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures". en. In: Leibniz-Institut für Deutsche Sprache. URL: https://hal.inria.fr/hal-02148693 (visited on 03/08/2023).

Sylvain Gugger Lysandre Debut, Thomas Wolf Philipp Schmid Zachary Mueller Sourab Mangrulkar (2022). *Accelerate: Training and inference at scale made simple, efficient and adaptable.* https://github.com/huggingface/accelerate.

Talafha, Bashar et al. (Dec. 2020). "Multi-dialect Arabic BERT for Country-level Dialect Identification". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. Barcelona, Spain (Online): Association for Computational Linguistics, pp. 111–118. URL: https://aclanthology.org/2020.wanlp-1.10.

Tan, Jiwei, Xiaojun Wan, and Jianguo Xiao (July 2017). "Abstractive Document Summarization with a Graph-Based Attentional Neural Model". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1171–1181. URL: https://aclanthology.org/P17-1108.

Tanfouri, Imen, Ghassen Tlik, and Fethi Jarray (2021). "An automatic arabic text summarization system based on genetic algorithms". In: *Procedia Computer Science* 189. AI in Computational Linguistics, pp. 195–202. ISSN: 1877-0509. URL: https://www.sciencedirect.com/science/article/pii/S187705092101200X.

Tang, P. C. et al. (2022a). "Otextsum: extractive text summarisation with optimal transport". In: *Findings of the Association for Computational Linguistics: NAACL 2022*.

Tang, P. C. et al. (2022b). "Tldw: extreme multimodal summarisation of news videos". In: URL: https://arxiv.org/abs/2210.08481.

Tay, Yi et al. (2023). *UL2: Unifying Language Learning Paradigms*. arXiv: 2205.05131 [cs.CL].

Touvron, Hugo et al. (2023a). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv: 2307.09288 [cs.CL].

Touvron, Hugo et al. (2023b). *LLaMA: Open and Efficient Foundation Language Models*. arXiv: 2302.13971 [cs.CL].

Tretyak, Vladislav and Denis Stepanov (2020). *Combination of abstractive and extractive approaches for summarization of long scientific texts*. arXiv: 2006.05354 [cs.CL].

Vaca, Alejandro et al. (June 2022). "Extractive and Abstractive Summarization Methods for Financial Narrative Summarization in English, Spanish and Greek". In: *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*. Marseille, France: European Language Resources Association, pp. 59–64. URL: https://aclanthology.org/2022.fnp-1.8 (visited on 03/06/2023).

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Vanetik, Natalia, Marina Litvak, and Sophie Krimberg (2022). "Summarization of financial reports with TIBER". In: *Machine Learning with Applications* 9, p. 100324. ISSN: 2666-8270. URL: https://www.sciencedirect.com/science/article/pii/S2666827022000391.

Vaswani, Ashish et al. (2017). *Attention Is All You Need.* arXiv: 1706.03762 [cs.CL].

Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh (June 2015). "CIDEr: Consensus-based image description evaluation". en. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, pp. 4566–4575. ISBN: 978-1-4673-6964-0. URL: http://ieeexplore.ieee.org/document/7299087/ (visited on 11/28/2021).

Verma, Rakesh M. and Daniel Lee (2017). "Extractive Summarization: Limits, Compression, Generalized Model and Heuristics". In: *Computación y Sistemas* 21.

Verma, Sukriti and Vagisha Nidhi (2019). *Extractive Summarization using Deep Learning.* arXiv: 1708.04439 [cs.CL].

Vhatkar, Amit, Pushpak Bhattacharyya, and Kavi Arya (Dec. 2020). "Knowledge Graph and Deep Neural Network for Extractive Text Summarization by Utilizing Triples". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 130–136. URL: https://aclanthology.org/2020.fnp-1.22 (visited on 10/16/2021).

Vig, Jesse (July 2019). "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, pp. 37–42. URL: https://www.aclweb.org/anthology/P19-3007.

Virtanen, Pauli et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17, pp. 261–272.

Wang, Ben and Aran Komatsuzaki (May 2021). *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.* https://github.com/kingoflolz/mesh-transformer-jax.

Wang, Danqing et al. (July 2020a). "Heterogeneous Graph Neural Networks for Extractive Document Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6209–6219. URL: https://aclanthology.org/2020.acl-main.553.

Wang, F. et al. (2023). "Self-augmentation improves zero-shot cross-lingual transfer". In: *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacifi*.

Wang, Haifeng et al. (2022). "Progress in Machine Translation". In: *Engineering* 18, pp. 143–153. ISSN: 2095-8099. URL: https://www.sciencedirect.com/science/article/pii/S2095809921002745.

Wang, Hong et al. (July 2019a). "Self-Supervised Learning for Contextualized Extractive Summarization". In: *Proceedings of the 57th Annual Meeting of*

the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, pp. 2221–2227. URL: https://aclanthology.org/P19-1214.
Wang, Kai, Xiaojun Quan, and Rui Wang (July 2019b). “BiSET: Bi-directional Selective Encoding with Template for Abstractive Summarization”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2153–2162. URL: https://aclanthology.org/P19-1207.
Wang, Lu et al. (2016). *A Sentence Compression Based Framework to Query-Focused Multi-Document Summarization*. arXiv: 1606.07548 [cs.CL].
Wang, Yu-An and Yun-Nung Chen (Nov. 2020b). “What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 6840–6849. URL: https://aclanthology.org/2020.emnlp-main.555.
Waskom, Michael et al. (Sept. 2017). *mwaskom/seaborn: v0.8.1 (September 2017)*. Version v0.8.1. URL: https://doi.org/10.5281/zenodo.883859.
Wenbo, Wang et al. (2019). *Concept Pointer Network for Abstractive Summarization*. arXiv: 1910.08486 [cs.CL].
Widyassari, Adhika Pramita et al. (2022). “Review of automatic text summarization techniques methods”. In: *Journal of King Saud University - Computer and Information Sciences* 34.4, pp. 1029–1046. ISSN: 1319-1578. URL: https://www.sciencedirect.com/science/article/pii/S1319157820303712.
Wikipedia contributors (2010). *Pearson correlation coefficient*. [Online; accessed 25-September-2023]. URL: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient/.
Wolf, Thomas et al. (Oct. 2020). “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6.
Workshop, BigScience et al. (2023). *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. arXiv: 2211.05100 [cs.CL].
Wu, Shijie et al. (2023). *BloombergGPT: A Large Language Model for Finance*. arXiv: 2303.17564 [cs.LG].
Wu, Shuangzhi et al. (July 2017). “Sequence-to-Dependency Neural Machine Translation”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 698–707. URL: https://aclanthology.org/P17-1065.
Wu, Yuxiang and Baotian Hu (2018). “Learning to Extract Coherent Summary via Deep Reinforcement Learning”. In: *AAAI Conference on Artificial Intelligence*.
Xiao, Min (July 2023). “Multi-doc Hybrid Summarization via Salient Representation Learning”. In: *Proceedings of the 61st Annual Meeting of the Association*

for Computational Linguistics (Volume 5: Industry Track). Toronto, Canada: Association for Computational Linguistics, pp. 379–389. URL: https://aclanthology.org/2023.acl-industry.37.

Xiao, Wen and Giuseppe Carenini (2019). "Extractive Summarization of Long Documents by Combining Global and Local Context". en. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3009–3019. URL: https://www.aclweb.org/anthology/D19-1298 (visited on 10/08/2021).

Xie, Niantao et al. (2018). "Abstractive Summarization Improved by WordNet-Based Extractive Sentences". In: *Natural Language Processing and Chinese Computing*. Ed. by Min Zhang et al. Cham: Springer International Publishing, pp. 404–415. ISBN: 978-3-319-99495-6.

Xu, Jiacheng and Greg Durrett (Nov. 2019). "Neural Extractive Text Summarization with Syntactic Compression". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3292–3303. URL: https://aclanthology.org/D19-1324.

Xue, Linting et al. (June 2021). "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 483–498. URL: https://aclanthology.org/2021.naacl-main.41.

Yadav, Chandra Shekhar (2018). *Automatic Text Document Summarization using Semantic-based Analysis*. arXiv: 1811.06567 [cs.IR].

Yadav, Deshraj et al. (2019). *EvalAI: Towards Better Evaluation Systems for AI Agents*. arXiv: 1902.03570 [cs.AI].

Yadav, Divakar, Jalpa Desai, and Arun Kumar Yadav (2022). *Automatic Text Summarization Methods: A Comprehensive Review*. arXiv: 2204.01849 [cs.CL].

Yang, Yi, Mark Christopher Siy UY, and Allen Huang (2020a). *FinBERT: A Pretrained Language Model for Financial Communications*. arXiv: 2006.08097.

Yang, Z. et al. (2020b). "Ted: a pretrained unsupervised summarization model with theme modeling and denoising". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Yao, Jin ge, Xiaojun Wan, and Jianguo Xiao (2015). "Compressive Document Summarization via Sparse Optimization". In: *International Joint Conference on Artificial Intelligence*.

Yen, Elizabeth H. (1968). "Elements of Nonparametric Statistics, Gottfried E. Noether, New York, John Wiley. Pp. X, 104." In: *Journal of the American Statistical Association* 63.322, pp. 728–728. eprint: https://doi.org/10.1080/01621459.1968.11009296. URL: https://doi.org/10.1080/01621459.1968.11009296.

Ying, Senci, Zheng Yan Zhao, and Wuhe Zou (2021). “LongSumm 2021: Session based automatic summarization model for scientific document”. en. In: *Proceedings of the Second Workshop on Scholarly Document Processing*. Online: Association for Computational Linguistics, pp. 97–102. URL: https://www.aclweb.org/anthology/2021.sdp-1.12 (visited on 10/08/2021).

Zabell, S. L. et al. (2008). “On Student’s 1908 Article ”The Probable Error of a Mean” [with Comments, Rejoinder]”. In: *Journal of the American Statistical Association* 103.481, pp. 1–20. ISSN: 01621459. URL: http://www.jstor.org/stable/27640017 (visited on 04/25/2023).

Zaheer, Manzil et al. (Jan. 2021). “Big Bird: Transformers for Longer Sequences”. In: *arXiv:2007.14062 [cs, stat]*. arXiv: 2007.14062. URL: http://arxiv.org/abs/2007.14062 (visited on 10/17/2021).

Zeng, Wenyuan et al. (2016). *Efficient Summarization with Read-Again and Copy Mechanism*. arXiv: 1611.03382 [cs.CL].

Zeroual, Imad et al. (Aug. 2019). “OSIAN: Open Source International Arabic News Corpus - Preparation and Integration into the CLARIN-infrastructure”. In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy: Association for Computational Linguistics, pp. 175–182. URL: https://aclanthology.org/W19-4619.

Zhang, Jianmin, Jiwei Tan, and Xiaojun Wan (2018a). *Towards a Neural Network Approach to Abstractive Multi-Document Summarization*. arXiv: 1804.09010 [cs.CL].

Zhang, Jingqing et al. (July 2020a). “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization”. In: *arXiv:1912.08777 [cs]*. arXiv: 1912.08777. URL: http://arxiv.org/abs/1912.08777 (visited on 03/01/2022).

Zhang, Susan et al. (2022). *OPT: Open Pre-trained Transformer Language Models*. arXiv: 2205.01068 [cs.CL].

Zhang, Tianyi et al. (Feb. 2020b). “BERTScore: Evaluating Text Generation with BERT”. In: *arXiv:1904.09675 [cs]*. arXiv: 1904.09675. URL: http://arxiv.org/abs/1904.09675 (visited on 11/22/2021).

Zhang*, Tianyi et al. (2020). “BERTScore: Evaluating Text Generation with BERT”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SkeHuCVFDr.

Zhang, Xingxing et al. (2018b). “Neural Latent Extractive Document Summarization”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 779–784. URL: https://aclanthology.org/D18-1088.

Zhao, Wei et al. (Nov. 2019). “MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 563–578. URL: https://aclanthology.org/D19-1053 (visited on 11/22/2021).

Zheng, Siyan, Anneliese Lu, and Claire Cardie (Dec. 2020). "SUMSUM@FNS-2020 Shared Task". In: *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. Barcelona, Spain (Online): COLING, pp. 148–152. URL: https://aclanthology.org/2020.fnp-1.25 (visited on 03/06/2023).

Zhong, Ming et al. (July 2019). "Searching for Effective Neural Extractive Summarization: What Works and What's Next". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1049–1058. URL: https://aclanthology.org/P19-1100.

Zhong, Ming et al. (July 2020). "Extractive Summarization as Text Matching". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6197–6208. URL: https://aclanthology.org/2020.acl-main.552.

Zhou, Qingyu, Furu Wei, and Ming Zhou (2020). *At Which Level Should We Extract? An Empirical Analysis on Extractive Document Summarization*. arXiv: 2004.02664 [cs.CL].

Zhou, Qingyu et al. (July 2017). "Selective Encoding for Abstractive Sentence Summarization". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1095–1104. URL: https://aclanthology.org/P17-1101.

Zhou, Qingyu et al. (July 2018). "Neural Document Summarization by Jointly Learning to Score and Select Sentences". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 654–663. URL: https://aclanthology.org/P18-1061.

Zhou, Yongxin, François Portet, and Fabien Ringeval (June 2022). "Effectiveness of French Language Models on Abstractive Dialogue Summarization Task". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3571–3581. URL: https://aclanthology.org/2022.lrec-1.382.

Zitouni, Imed (2011). *Introduction to Arabic Natural Language Processing.*

Zmandar, Nadhem, Mahmoud El-Haj, and Paul Rayson (2021a). "Multilingual Financial Word Embeddings for Arabic, English and French". In: *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4584–4589.

— (2023a). "A Comparative Study of Evaluation Metrics for Long-Document Financial Narrative Summarization with Transformers". In: *Natural Language Processing and Information Systems*. Ed. by Elisabeth Métais et al. Cham: Springer Nature Switzerland, pp. 391–403. ISBN: 978-3-031-35320-8.

Zmandar, Nadhem, Mo El-Haj, and Paul Rayson (2023b). "FinAraT5: A text to text model for financial Arabic text understanding and generation." In: *4th Conference on Language, Data and Knowledge*.

Zmandar, Nadhem et al. (2021b). "Joint abstractive and extractive method for long financial document summarization". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 99–105.

Zmandar, Nadhem et al. (2021c). "The Financial Narrative Summarisation Shared Task FNS 2021". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 120–125. URL: https://aclanthology.org/2021.fnp-1.22 (visited on 03/06/2023).

Zmandar, Nadhem et al. (2021d). "The Financial Narrative Summarisation Shared Task FNS 2021". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*, pp. 120–125.

Zmandar, Nadhem et al. (2021e). "The Financial Narrative Summarisation Shared Task FNS 2021". In: *Proceedings of the 3rd Financial Narrative Processing Workshop*. Lancaster, United Kingdom: Association for Computational Linguistics, pp. 120–125.

Zmandar, Nadhem et al. (June 2022). "CoFiF Plus: A French Financial Narrative Summarisation Corpus". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 1622–1639.

Çelikyılmaz, A. et al. (2018). "Deep communicating agents for abstractive summarization". In: *Proceedings of the 2018 Conference of the North American Chapter Of the Association for Computational Linguistics: Hu*.