Multi-document Arabic Text Summarisation

Mahmoud El-Haj

A thesis submitted for the degree of

Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

2012

Abstract

Multi-document summarisation is the process of producing a single summary of a collection of related documents. Much of the current work on multi-document text summarisation is concerned with the English language; relevant resources are numerous and readily available. These resources include human generated (gold-standard) and automatic summaries. Arabic multi-document summarisation is still in its infancy. One of the obstacles to progress is the limited availability of Arabic resources to support this research. When we started our research there were no publicly available Arabic multi-document gold-standard summaries, which are needed to automatically evaluate system generated summaries. The Document Understanding Conference (DUC) and Text Analysis Conference (TAC) at that time provided resources such as gold-standard extractive and abstractive summaries (both human and system generated) that were only available in English. Our aim was to push forward the state-of-the-art in Arabic multi-document summarisation. This required advancements in at least two areas. The first area was the creation of Arabic test collections. The second area was concerned with the actual summarisation process to find methods that improve the quality of Arabic summaries. To address both points we created single and multi-document Arabic test collections both automatically and manually using a commonly used English dataset and by having human participants. We developed extractive language dependent and language independent single and multi-document summarisers, both for Arabic and English. In our work we provided state-of-the-art approaches for Ara-

bic multi-document summarisation. We succeeded in including Arabic in one of the leading summarisation conferences the Text Analysis Conference (TAC). Researchers on Arabic multi-document summarisation now have resources and tools that can be used to advance the research in this field.

Acknowledgement

Writing this doctoral thesis would not have been possible without the help and continuous support from the people around me during this wonderful journey, to only some of whom it is possible to give particular mention here. I would like to thank my parents, brother and sisters for their love and support, and for believing in me and my goals.

This thesis would not have seen the light without the enormous and endless support, patience and guidance of my supervisors, Dr Udo Kruschwitz and Dr Chris Fox, for which my mere expression of thanks likewise does not suffice. I was always amazed with how Dr Kruschwitz and Dr Fox can split their time equally between their students and the way they break any problem into a very few simple steps. I have been to many talks by Dr Kruschwitz and in each one of them he did not once forget to mention his students and praise the work they have done. Having Dr Kruschwitz and Dr Fox as my supervisors is a great privilege.

My time as a PhD student at Essex University was great with lots of wonderful memories. By the time I started my PhD I was a bit worried of losing my social life but luckily that was not the case. I was able to split my time between my PhD, my part-time job, playing football, travelling and connecting with my friends.

Last but not the least, I would like to thank my fellow PhD students and my friends at Essex University. They each helped make my time during this PhD program more fun and interesting.

Contents

List of Figures

List of Tables

Glossary

Notation	Description
Natural Language Processing (NLP)	the branch of information science that deals with natural language information.
Natural Language Generation (NLG)	the natural language processing task of generating natural language from a machine representation system.
Information retrieval (IR)	the science of searching for documents or information within documents.
Information extraction (IE)	a type of information retrieval whose goal is to automatically extract structured information from unstructured documents.
Automatic Summarisation	the creation of a shortened version of a text by a computer program.
Sentence Compression	the task of producing a summary of a single sentence.
Abstractive Summarisation	using NLG to generate a system summary.
Extractive Summarisation	using IE to generate a system summary.
Query-based Summary	a summary that presents the contents of a document that are related to a user's query.
Generic-based Summary	a summary that presents an overall sense of a documents' contents.
Clustering	the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar to each other than to those in other clusters.
Cluster	a group of similar objects growing closely together.
Machine Learning	a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data.
Relevance Score	how relevant a search result is, based on the search terms.

Notation	Description
Hidden Markov Model (HMM)	is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved state.
Supervised learning	a machine learning task of inferring a function from supervised (labelled) training data.
Unsupervised learning	a machine learning task of inferring a function from unlabelled data.
Latent Semantic Indexing/Analysis	is an indexing and retrieval method that uses a mathematical techniques to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text.
Semantic Analysis	a phase of NLP that involves extraction of context-independent aspects of a sentence's meaning.
Syntactic Analysis (Parsing)	is the process of analysing a text, made of a sequence of tokens to determine its grammatical structure with respect to a given formal grammar.
Statistical Analysis	a collection of methods used to process large amounts of data and report overall trends.

Chapter 1

Introduction

1.1 Motivation

The volume of information available on the Web is increasing rapidly with the rise in numbers of Internet users. According to the Internet World Stats[1], the number of Internet users exceeded two billion by the end of 2011. The need for systems that can automatically summarise documents is becoming ever more desirable. For this reason, text summarisation has quickly grown into a major research area as illustrated by the Text Analysis Conference (TAC) and the Document Understanding Conference (DUC) series, which started in 2001. We are interested in the automatic summarisation of Arabic documents. Research in Arabic is receiving growing attention [Yaseen and Theophilopoulos, 2001], [Douzidia and Lapalme, 2004], [Haddad and Yaseen, 2005], [Hadrich et al., 2005], [Khreisat, 2006], [Benmamoun, 2007], [Diab et al., 2007], [Benmamoun, 2007], [Al-Shammari and Lin, 2008], [Schlesinger et al., 2008], [Alghamdi et al., 2009], [Salhi, 2010], [Habash and Roth, 2011] and [Diehl et al., 2012]. It has been widely acknowledged that apart from a few notable exceptions such as the Arabic

[1] http://www.internetworldstats.com/stats.htm

Penn Treebank[1] and the Prague Arabic Dependency Treebank[2] there are few publicly available tools and resources for Arabic natural language processing (NLP). Arabic NLP lacks resources such as Arabic corpora, lexicons, machine–readable dictionaries in addition to fully automated fundamental NLP tools such as tokenizers, part of speech taggers, parsers, and semantic role labelers [Diab et al., 2007]. However, this has started to change in recent years [Alghamdi et al., 2009; Diehl et al., 2012; Habash and Roth, 2011; Maegaard et al., 2008]. Some reasons for this lack of resources and tools may be due to the complex morphology, the absence of diacritics (vowels) in written text and the fact that Arabic does not use capitalisation, which causes problems for named entity recognition [Benajiba et al., 2009]. Tools and resources however are essential to advance research in Arabic NLP. In the case of automatic summarisation tasks, most of the activities are concerned with the English language, as with TAC and DUC conferences. This focus is reflected in the availability of resources: in particular, there were no readily available gold standards for evaluating Arabic summarisers by the time we started this research. Tools and resources are essential to advance research in Arabic NLP, but generating them with traditional techniques is both costly and time-consuming.

1.2 Statement of the Problem

As researchers, we would be interested in a system that can summarise a set of related work by providing a short summary with the work that has been done on a certain field, or a summary for a set of related articles. This is just a simple example of the facilities that a summariser system could provide. Another example would be having a system that can answer a query using a clustered set of related documents and provide a summery for each cluster. The idea behind summarising a set of documents with one

[1] http://www.ircs.upenn.edu/arabic/
[2] http://ufal.mff.cuni.cz/padt/PADT 1.0/

single summary could save users' time and effort. While many systems focusing on English and European languages have been developed, the field of Arabic summarisation still needs more attention. Search engines such as Bing[1] provide the user with a short summary for each of the search results by just hovering the mouse over the article. To illustrate an example for what we intend to develop, we could imagine someone using a web search engine to retrieve articles about work related to text summarisation in Arabic. Instead of visiting each of the resulting pages, the user would be interested in a short summary, not just a single summary for each article, but a summary for a cluster of related articles. For example, one cluster could contain a set of articles talking about single document summarisation while another contains those articles related to summarisation using machine learning, now the user can click or hover on each cluster to summarise it. Figures 1.1 and 1.2 show examples on summarising a set of related documents. The first example, Figure 1.1, illustrates an abstract idea of summarising a number of n related articles written in Arabic. The documents talk about the closing of the 2008 Summer Olympic games in China, the parallel English translation can be found in the next example in Figure 1.2.

1.3 Area of Study

The main goal of this research is to develop techniques for multi-document Arabic text summarisation that can summarise a set of related text documents written in the Arabic language. A successful summarisation approach needs a good guide to find the most important sentences that are relevant to a certain criteria. Therefore, the proposed methods should work on extracting the most important sentences from a set of related articles. To create those methods and techniques we apply Arabic natural language processing tools in addition to statistical and semantic models.

[1] http://www.bing.com

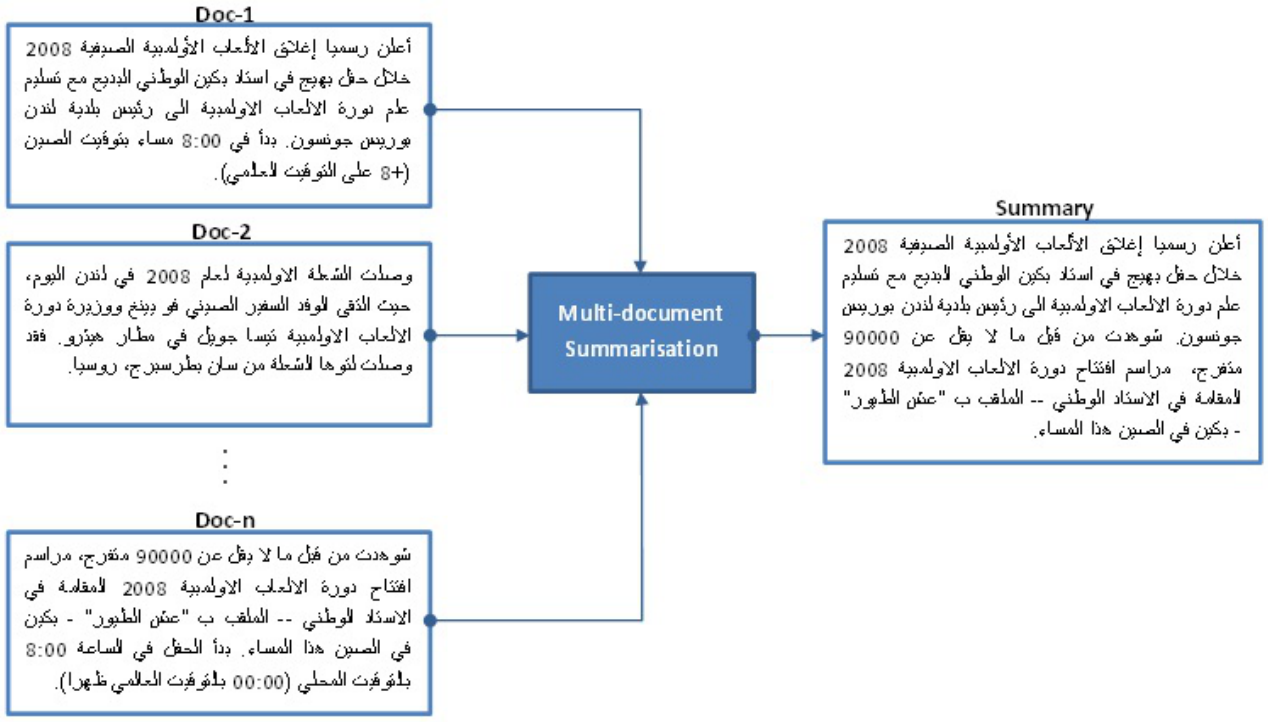

Figure 1.1: Summarising multi-documents in Arabic

1.4 Research Questions

The work proposed here attempts to address the following research questions:

1. What are the most effective methods for Arabic multi-document text summarisation?

2. Can single document text summarisation methods be effectively applied to multi-document summarisation?

3. What are the language-independent and the specific aspects of the Arabic language affecting the quality of multi-document text summarisation?

4. How do the developed methods compare to state-of-the-art technologies when run on standard test collections?

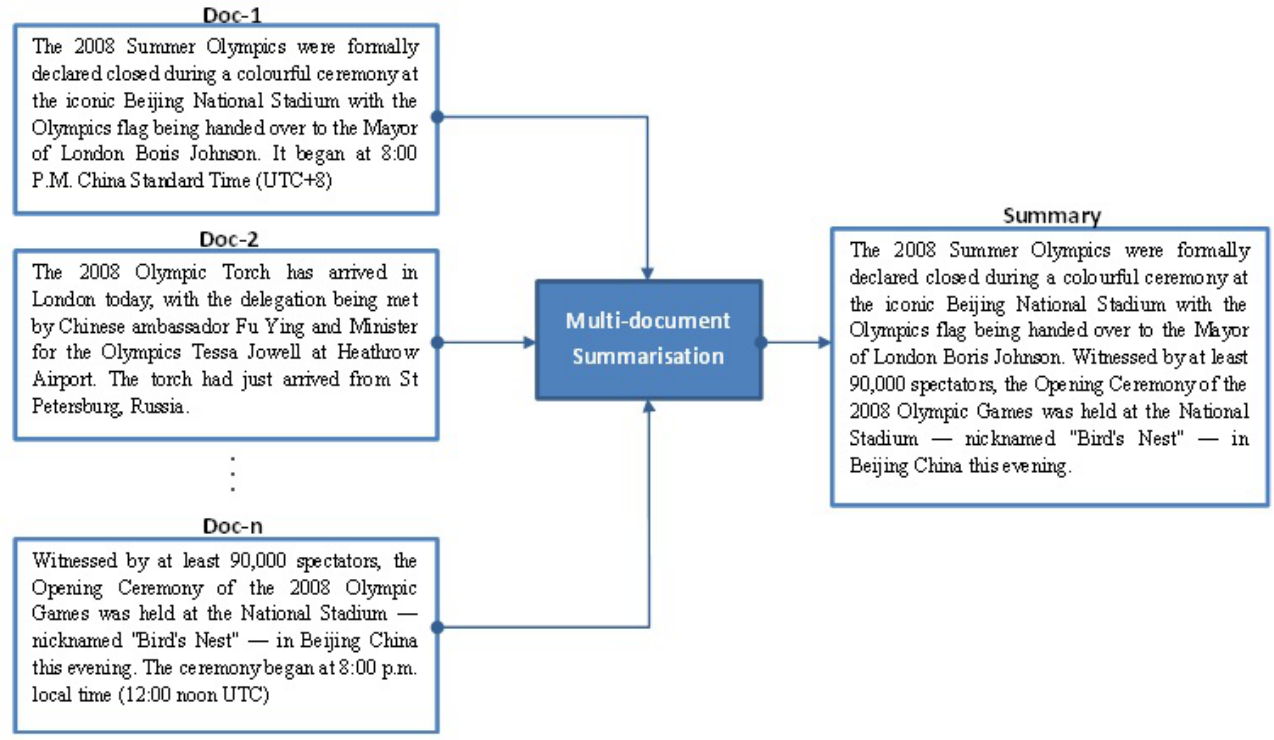

Figure 1.2: Summarising multi-documents in English

1.5 Contributions

The research described here accomplished several goals. In this section we list the contributions, later on we discuss them in detail.

- We created a human-generated corpus of extractive Arabic summaries for a selection of Arabic articles and newswires using an online workforce.
- We created an Arabic multi-document summaries corpus by using a machine translation tool to translate articles from English into Arabic.
- We developed extractive language–dependent and language–independent single and multi-document summarisers, both for Arabic and English.
- We created a multi-document Arabic corpus and multi-document gold-standard human summaries, which is a valuable step that could advance the research on

Arabic multi-document summarisation.

- We succeeded in including Arabic in one of the leading summarisation conferences the Text Analysis Conference (TAC). Being part of the organising committee for the MultiLing (multi-lingual) summarisation pilot, at the same conference, we were responsible for creating and providing Arabic resources, which included human and system generated multi-document summaries in addition to a human translated corpus.

- We provided state-of-the-art approaches for Arabic multi-document summarisation and redundancy elimination. Researchers on Arabic multi-document summarisation now have resources and tools that can be used to advance the research in this field.

1.6 Organisation of Thesis

The organisation of the rest of the thesis is as follows. Chapter 2 presents the detailed background of the different summarisation methods and techniques and illustrates the key area of related work, the chapter also gives a detailed background on Arabic language and processing tools. Chapter 3 presents a framework for automatic text summarisation and talks about the general architecture and the process of creating summarisation corpora in addition to the general breakdown of creating a summariser. The chapter shows the contribution made to Arabic single and multi-document summarisation in addition to the summaries methodologies. Chapter 4 illustrates the work being done to create resources for Arabic single and multi-document summarisation, which include the creation of Arabic gold-standard summaries, both human and automatic generated summaries. Chapter 5 presents methods and implementations of Arabic extractive single-document and multi-document summarisers, the summarisers

description and the experimental setup. Chapter 6 shows evaluation results of our extractive single-document summarisation approaches. The chapter addresses the effect of the summary-length on the evaluation scores. Chapter 7 shows the evaluation results of the work done on multi-document. The chapter also shows the evaluation results of the experiments compared against other summarisation techniques and systems from the literature. We finish with conclusions and future work in Chapter 8.

This thesis includes work published elsewhere [Aker. et al., 2012; El-Haj et al., 2009, 2010, 2011a,b,c; Giannakopoulos et al., 2011].

Chapter 2

Related Work

In human summarisation one tends to summarise a single article by summing up the most important ideas and ordering to ensure they are coherent. Even for humans this task would take a lot of work. Summaries produced by two different people would typically be different. Different individuals may have a different view about what is important. This inspired the need for having an automatic summariser that can perform the job in less time and with the least effort. This led for the research on automatic summarisation to start more than 50 years ago [Luhn, 1958].

Text summarisation in general is the process of summarising a single article or a set of related ones by summing up the most important events, making sure the events sequence is coherent by ordering them chronically. On the other hand automatic text summarisation is the process of producing a shortened version of a text by the use of computers. For example reducing a text document or a group of related documents into a shorter version of sentences or paragraphs using automated tools and techniques. The summary should convey the key contributions of the text. In other words, only key sentences should appear in the summary and the process of defining those sentences is highly dependant on the summarisation method used.

In automatic summarisation there are two main approaches that are broadly used,

extractive and abstractive. The first method, the extractive summarisation, extracts, up to a certain limit, the key sentences or paragraphs from the text and orders them in a way that will produce a coherent summary. The extracted units differ from one summariser to another. Most summarisers use sentences rather than larger units such as paragraphs. Extractive summarisation methods are the focus method on automatic text summarisation. The other method, abstractive summarisation, involves more language dependent tools and Natural Language Generation (NLG) technology. Such summarisers can include words not present in the original document. The idea of abstractive summarisation seeks to mimic the human summarisation methods, but it is much harder to implement. In our research we adopted the extractive summarisation approach. Work on automatic summarisation dates back more than 50 years, with a focus on the English language [Luhn, 1958]. The work on Arabic automatic summarisation is more recent and still not on par with the research on English and other European languages. Early work on Arabic summarisation started less than 10 years ago [Conroy et al., 2006; Douzidia and Lapalme, 2004]. There has been a shortage of Arabic resources needed to advance the work in this field, and the acquisition of the appropriate resources to push the state-of-the-art in Arabic text summarisation represents the contributions of the work described in this research [El-Haj et al., 2010, 2011a,c; Giannakopoulos et al., 2011]. In the rest of this chapter we will discuss related work on automatic summarisation for the general approaches and methods mentioned above.

2.1 Types of Summarisation

When looking at text summarisation in detail, we can distinguish a range of different dimensions. Over time, there have been various approaches to automatic text summarisation. These approaches include single-document and multi-document sum-

marisation. Both single-document and multi-document summarisation use the summarisation methods mentioned earlier, i.e. extractive or abstractive summarisation. Summarising a text could be dependent on input information such as a user query or it could be *generic* where no user query is used. Figure 2.1 illustrates the different automatic summarisation approaches, methods and techniques.

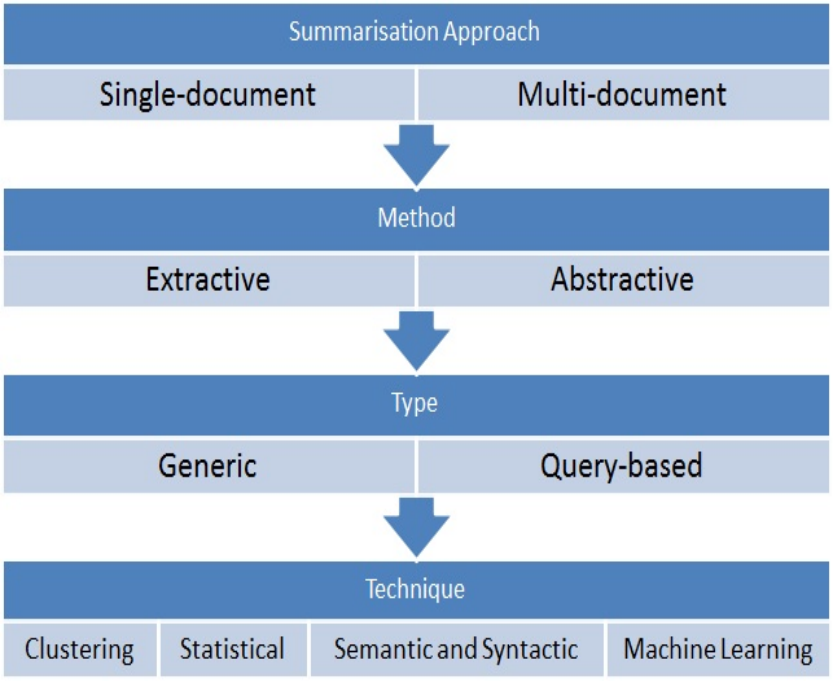

Figure 2.1: Summarisation Approaches Diagram

As shown in Figure 2.1, query-based (query-focused) summarisation works for both single and multi-document summarisation, it is the process of summarising a document based on the user's information need. The process of query-based summarisation is mainly about retrieving sentences that match a certain user query [Bawakid and Oussalah, 2008; Chen and Verma, 2006; Chowdary and Kumar, 2008; Nastase et al., 2008; Qiu and Frei, 1993; Ying et al., 2007]. In our work most of the summarisation approaches we propose are generic; there is no query.

The length of the generated summary can vary and will depend on the purpose of the summarisation process. Task descriptions of the summarisation tracks organ-

ised by the Document Understanding Conference (DUC) and later on by the Text Analysis Conference (TAC) specified the summary length to be within a certain limit of words (e.g., between 240 and 250 words inclusive, white-space-delimited tokens). Summaries over the size limit were truncated and summaries below the size limit were penalised by reducing its score during the evaluation process. The compression ratio (how much shorter the summary is than the original) in other summarisers varies from short summaries (100 words) [Angheluta et al., 2004] to very short summaries (ten words) [Douzidia and Lapalme, 2004]. In our work we want the summary length to be flexible by providing a scalable compression ratio, which is dependent on the extraction technique and the summarisation approach used. In Chapter 6 we study the effect of the summary length on the automatic evaluation scores.

Extraction tends to play an important role in single-document and multi-document summarisation. Assessing the importance of the extracted units typically depends on some statistical measures. Each unit is given a score based on features such as word frequencies [Luhn, 1958], position in the text [Baxendale, 1958], and the presence of key phrases [Edmundson, 1969]. Other approaches use more sophisticated techniques for deciding which sentences to extract. These techniques include machine learning (e.g., [Leite and Rino, 2008]) to identify important features, and various natural language processing techniques to identify key passages and relationships between words. Bayesian classifiers have also been used [Kupiec et al., 1995]. The work by [Fung and Ngai, 2006] used Hidden Markov Models (HMMs) to reflect the fact that the probability of inclusion of a sentence in an extract depends on whether the previous sentence has also been included.

Figure 2.2 illustrates the most common summarisation techniques and methods that will be discussed throughout the related work section. The diagram is illustrative; the suggested distribution is not strict due to the large overlap between automatic text summarisation approaches.

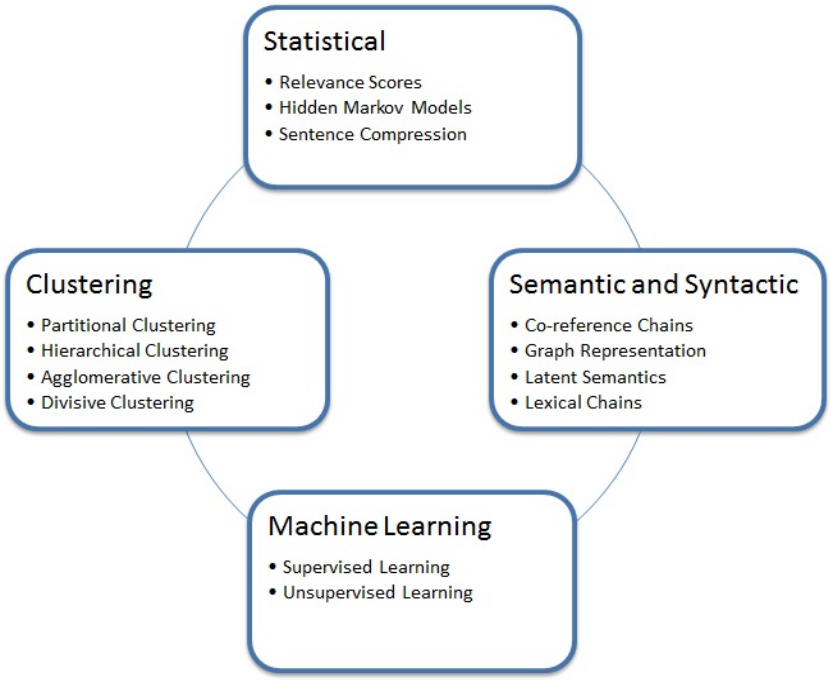

Figure 2.2: Summarisation Techniques Diagram (with examples)

Let us turn to single-document summarisation. The idea of single-document summarisation varies from a basic architecture using a simple extraction process to more complicated ones using deep semantic processing. The approach of single-document summarisation relies on the idea of producing a summary for a single document. The main factor in single-document summarisation is to identify the most important (informative) parts of a document. Early work on single-document summarisation was the work by [Luhn, 1958]. In his work he looked for sentences containing keywords that are most frequent in a text. The sentences with highly weighted keywords were selected. The work by Luhn [1958] highlighted the need for features that reflect the importance of a certain sentence in a text. Baxendale [1958] showed the importance of sentence-position in a text, which is understood to be one of the earliest extracted features in automatic text summarisation. They took a sample of 200 paragraphs and found that in 80% of the paragraphs the most important sentence was the first one.

Multi-document summarisation produces a single summary of a set of documents.

The documents are assumed to be about the same genre and topic. The analysis in this area is performed typically at either the sentence or document level. Information Extraction (IE) was used in an early approach to multi-document summarisation to find similarities and differences between sentences in documents [McKeown and Radev, 1995; White and Cardie, 2002]. Funk et al. [2007] in later work merged IE with a process that regenerates the extracted units in order to improve the summarisation technique. As we will see in the related work below, most of the work on single-document and multi-document summarisation was done using the extractive summarisation method, where the most important information from a document or a set of related ones are extracted as units and combined to generate a summary. As in single-document summarisation, extractive-based approaches were also applied on multi-document summarisation. Goldstein et al. [2000] discussed a text extraction approach for multi-document summarisation. It was built on single-document summarisation methods using available information about the document set as a whole and the relationships between the documents using domain independent techniques. The idea of selecting the sentences that hold the key contributions of the text is common between most of the summarisation approaches, different tools and techniques were applied to try to enhance the selection process. Lin and Hovy [2002] introduced a multi-document summarisation system called NeATS which attempted to extract relevant portions from a set of related documents and presented them in coherent order.

So what are the distinguishing factors between single and multi-document summarisation apart from the input sources? Redundancy elimination is one of the main differences between single and multi-document summarisation. Selecting sentences from a set of related articles could result in overlapping information that is considered redundant and must be eliminated. Semantic, syntactic and statistical models have been used to eliminate redundant sentences in multi-document summarisation. Fukumoto et al. [2010] used clustering to eliminate redundancy where they classified

the extracted sentences into groups of semantically related sentences. Their summarisation approach focused on detecting key sentences, that contain crucial information, from related documents. Following the same idea of detecting semantically related sentences Hendrickx et al. [2009] used a semantic overlap detection tool to identify redundant sentences in their Dutch multi-document summariser system. Many tools and techniques have been investigated to detect redundant sentences. Newman et al. [2004] described an experiment to determine the quality of different similarity metrics, with regard to redundancy elimination. The three metrics examined were WordNet distance (using a semantic lexicon for the English language), Cosine Similarity (a measure of similarity between two vectors by measuring the cosine of the angle between them) and Latent Semantic Indexing/Analysis (LSI/LSA) [Deerwester et al., 1990]. Hirao et al. [2004] proposed a multi-document summarisation system, which employed a sequential pattern mining algorithm for sentence extraction and the Maximum Marginal Relevance (MMR) [Carbonell and Goldstein, 1998] to minimise the redundancy of the extracted sentences. Hachey et al. [2005] used a latent semantic space to model sentence similarity over a large corpus for detecting and eliminating redundancy. Bossard and Rodrigues [2010] combined multi-document summarisation with a genetic algorithm [Banzhaf et al., 1998]. They proposed a summarisation system called CBSEAS, which integrated clustering to detect redundant sentences in order to generate high quality summaries. A genetic algorithm was used to adapt the system to specific domains. In our work we will also use different methods, techniques and models to eliminate redundant sentences. The main difference is that we apply those techniques on Arabic multi-document summarisers.

Many approaches and techniques have been investigated, which show that the research on single-document and multi-document summarisation is a growing research area and many methods, models and techniques have been introduced. There is a large overlap between the summarisation methods and approaches. Most of the approaches

proposed use the extraction method for the summarisation process. The remaining sections will further discuss the related work by focusing on the various techniques, models and tools used in automatic text summarisation.

2.1.1 Semantic-based and Syntactic–based Summarisation

Text summarisation based on semantic analysis and semantic correlations between sentences has been used extensively, e.g., [Blair-Goldensohn and McKeown, 2006; Hovy and Lin, 1996; Huang et al., 2006; Toutanova et al., 2007]. Many semantic analysis techniques have been applied on text summarisation to find relations between sentences. Figure 2.3 shows some of these techniques, which include textual entailment [Lacatusu et al., 2006] and graph representation using lexical graphs [Boudin et al., 2007; Filippova et al., 2007; Günes, 2006; Witte et al., 2007]. Other techniques include semantic clustering, co-reference and lexical chains, lexical semantic and anaphoric resolution [Azzam et al., 1999; He et al., 2008; Huang et al., 2006; Vanderwende et al., 2006; Zhou et al., 2005] in addition to latent semantic analysis (LSA) [Hachey et al., 2005]. The diagram is illustrative; the suggested distribution is not strict due to the large overlap between semantic and syntactic approaches.

Azzam et al. [1999] used co-reference chains to generate text summaries. They used a variety of criteria to select the *best* chain to represent the main topic of a text. The summary representation is a *best chain*, which is selected from the set of co-reference chains by the application of one or more heuristics. The summary provided by the system is simply the fusion of sentences from the original text which contain one or more expressions occurring in the selected co-reference chain. Similarly, Zhou et al. [2005] applied lexical chains in their summariser *IS–SUM*. They integrated Document Index Graphics (DIG) [Hammouda and Kamel, 2004] with lexical chains, which helped them in summarising long documents efficiently. They made some modifications to increase

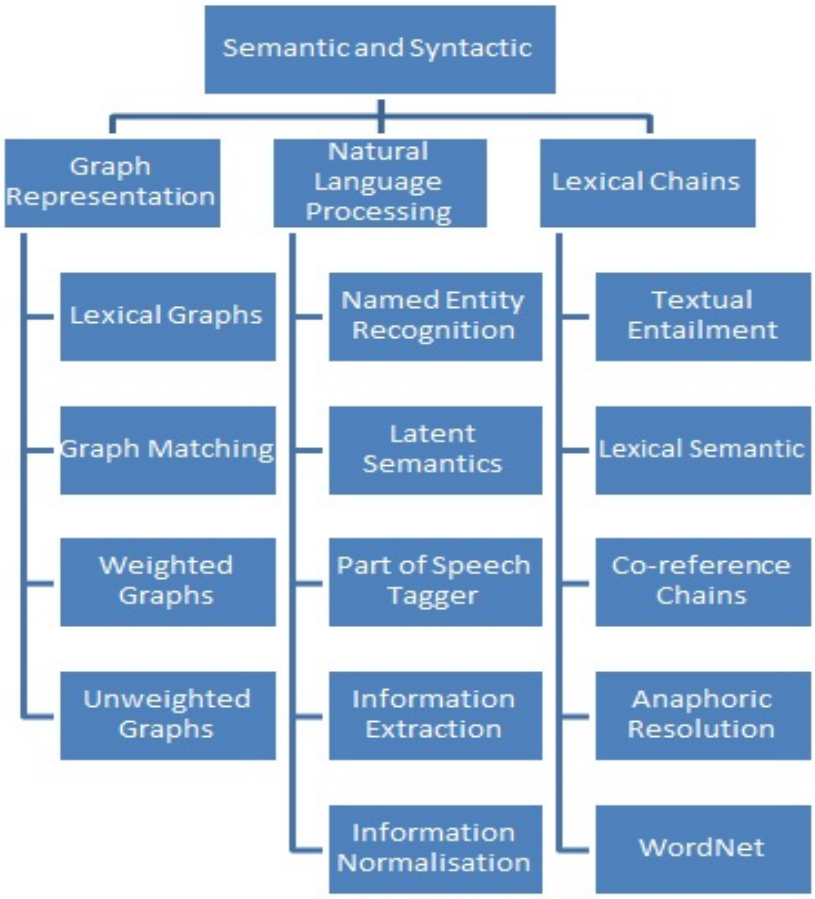

Figure 2.3: Semantic and Syntactic Techniques

the precision and recall ratio of the summarisation output afterwards. Improvements to the semantic and syntactic analysis fields led to improvements on the summariser's quality.

WordNet Lexical databases such as WordNet[Fellbaum, 1998] have been used in automatic text summarisation to find semantic relationship between sentences. WordNet groups words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets.Li et al. [2006] enhanced the *IS–SUM* summariser system by improving its lexical chain algorithm through applying WordNet for similarity calculation. Following the same approach, Filippova et al. [2007] presented a summariser system that combines document filter-

ing, ranking sentences using lexical chains and graph matching algorithms. The system takes the top 30 most relevant sentences selected by the lexical chaining method as the input to another ranking algorithm which re-ranks presumably relevant sentences according to a different criterion to generate the final summary.

Bawakid and Oussalah [2008] proposed a methodology for query-based extractive summarisation. The methodology used phrasal decomposition of text where each sentence was assigned a relevance score, which was used to identify the most relevant sentences in the documents. The scoring function includes a semantic similarity evaluation where WordNet was used in conjunction with a variety of other extracted features to construct the sentence-to-sentence semantic similarity. Similarly, Verma et al. [2007] proposed a query-based text summarisation technique that made use of WordNet. The summarisation system was tuned to summarise medical documents by integrating the Unified Medical Language System[1], a medical ontology knowledge source from the National Library of Medicine.

As illustrated in Figure 2.3, other levels of semantic and syntactic representation have been investigated. Lexical similarity (lexical semantic) was proposed by Flores et al. [2008], they presented *LIC2M*, a semantic strategy for summarising documents through extraction. They used a "bag of senses" to calculate sense concentration on each sentence. Text sentences are ranked according to their lexical similarity against a topic statement represented as a bag of words. Blake et al. [2007] described the *UNC-CH* system. The proposed system generated a topic-focused summary of information reported in multiple news articles. They explored query expansion, lexical simplification and sentence simplification to improve summarisation performance.

Figure 2.3 also shows the range of natural language processing tools that have been used to enhance the summarisation process. Han et al. [2004] presented the KU multi-document summarisation system. The summariser used lexical and syntactic

[1] `http://www.nlm.nih.gov/research/umls/`

tools to extract information from documents to generate summaries. Natural language processing tools used in their summarisers included information extraction, information normalisation, information fusion, and summary generation modules. Zhao et al. [2009] explored a method for a query-focused multi-document summarisation by using sentence-to-sentence and sentence-to-word relations to select the query words from the documents set. Those relations were used as query expansion to enhance the sentence ranking process. Stokes et al. [2007] proposed an extractive summarisation approach that used query expansion to choose sentences to generate a 250–word summary. They calculated the cosine similarity between each sentence and an expanded form of the query. The original query is used to retrieve an initial set of relevant documents, and then the top 20 most frequent words in this ranked list are extracted and used to expand the original query. Chen and Verma [2006] proposed a user query based text summarisation for medical literature. They used ontological knowledge to find relations between terms in the medical domain to enhance the summarisation process. Schiffman [2007] presented an experimental system that included contextual information in query-focused summarisation. In addition, the system incorporated corpus-driven semantic information to find semantic relationships between passages and the query. The passages that are most similar to the query were selected to be in the summary.

As illustrated in Figure 2.3, graph representation, weighted graph and unweighted [Bollobas, 1998] were applied on semantic and syntactic summarisation. Günes [2006] introduced an algorithm that represented the set of sentences in a document as a graph. The nodes of the graph were sentences, while the links between the nodes were induced by a similarity relation between the sentences. Sentences were ranked according to a random walk model defined in terms of both the inter-sentence similarities and the similarities of the sentences to the topic description. Boudin et al. [2007] presented the *LIA1* summarisation systems. The system proposed was a combination of seven different sentence selection systems. The fusion of the system outputs was made with

a weighted graph where the cost functions integrate the votes of each system. The final summary corresponds to the best path in this graph. As for the graph representation Giannakopoulos et al. [2008] used the n-gram graph representation to offer rich information on the contextual relations between character or word n-grams throughout the whole pipeline of the proposed summarisation system.

Tasks, tracks and datasets by leading summarisation conferences such as the Text Analysis Conference (TAC) and the Document Understanding Conference (DUC) have contributed to advancing the research on text summarisation aspects, especially for update and opinion-focused summarisation. Witte et al. [2007] used "fuzzy co-reference cluster graphs" to generate update summaries. Their technique was based on grouping noun phrases into fuzzy co-reference chains. They generated the cluster graph data structure based on context containing a single or set of questions and a number of document clusters containing topical documents sorted by their publication date. The work was part of the DUC 2007 update summary pilot, where participants had to create short (100-word) multi-document summaries. Seki et al. [2006] presented an opinion-focused extractive summarisation approach to build a multi-document summariser on the basis of user-specified summary viewpoints at the DUC 2006 task. In their approach sentence type annotation was used for weighting, the annotations define sentiments which included opinionated, positive, and negative sentence types. Frequencies of terms were also taken into account when weighting sentences. Wang and Li [2011] proposed a weighted consensus summarisation method to combine results from single-document summarisation systems. They studied different methods for multi-document summarisation, including centroid-based, graph-based, and dimension reduction to improve the summarisation quality. They used the DUC–2002 and DUC–2004 data sets. They compared their method with various baselines based on average score, average rank and other combinations. Information retrieval was one of the many techniques presented.

Semantic models and language-specific techniques have been used frequently. Semantic models are conceptual data models in which semantic information is included. Hovy and Lin [1996] developed a multilingual text summarisation system named SUMMARIST. This included language-specific techniques of parsing and semantic analysis, and combined robust NLP processing. D'Avanzo and Magnini [2005] introduced a key-phrase extraction methodology, named LAKE, to identify relevant terms in the document. In their work more sophisticated techniques were presented such as a Part Of Speech (POS) tagger (to decide if a given word is a noun, verb, adjective, etc) and Named Entities Recognition (NER) to process a text and identify expressions that refer to people, places, companies and so on. A score mechanism based on the frequency of the key-phrases was used to score the best sentences in documents. Li et al. [2006] presented a query-based multi-document summarisation system. It was an extended version of a generic multi-document summarisation system named *PoluS 1.0*. The system incorporated Latent Semantic Analysis (LSA) technology to make the generated summaries satisfy the user's information need. Conroy and Schlesinger [2008] used Latent Semantic Indexing (LSI) to remove redundant sentences and generate a 250 word summary consisting of complete sentences.

The relevance of the extracted sentences in extractive summarisers play a big role in the quality of the generated summaries. Lacatusu et al. [2006] showed that by considering entailment relationships between sentences extracted for a summary, they could automatically create semantic "Pyramids" used to identify answer passages that are both relevant and responsive. Blair-Goldensohn and McKeown [2006] added a semantically-motivated aspect to their work on summarisation by integrating models of rhetorical-semantic phenomena such as causality and contrast. Huang et al. [2006] presented a method to extract key sentences of a document as a summary by estimating the relevance of sentences by using fuzzy-rough sets. This method used senses rather than raw words so that sentences of the same or similar semantic are treated differently.

Semantic clustering has been used to avoid selecting redundant key sentences. He et al. [2008] used syntactic-based anaphora resolution and sentence compression algorithms. Term significance was then obtained by frequency-related topic significance and query-related significance by obtaining co-occurrence information with query terms. Barrera and Verma [2011] used a combination of syntactic and semantic approaches. They introduced *SYNSEM*, which is a single-document summariser that exploits a document's word popularity, sentence position, and semantic linkage as three main approaches for sentence extraction.

Despite all the progress in text summarisation, Arabic semantic and syntactic summarisation is still in its early stages. Only a few techniques and systems exists. Schlesinger et al. [2008] tackled the field of semantic and syntactic Arabic summarisation through their multi-document summariser so called *CLASSY*. *CLASSY* (Clustering, Linguistics, And Statistics for Summarisation Yield) is an automatic, extract-generating, summarisation system that uses linguistic trimming and statistical methods. The Arabic dataset used was translated from English using an automatic machine translation tool. The quality of the generated summaries was affected by the poor performance of the translation process. Later on, Azmi and Al-Thanyyan [2012] presented an automatic extractive Arabic text summarisation system where the user can cap the size of the final summary. To generate summaries they used Rhetorical Structure Theory (RST) [Mann and Thompson, 1988] – an approach to text structure to characterise text and textual relations for the purpose of text generation. They assigned a score to each of the sentences in the primary summary. These scores helped in generating the final summary.

Concluding Remarks

Many summarisation systems use syntactic, semantic and some forms of deep language processing for their text summarisers. The use of natural language processing tools with semantic and syntactic models in automatic text summarisation has contributed to increasing the quality of the generated summaries, which was achieved by suggesting better sentence-to-sentence relationships. The related work showed some progress on tackling the field of Arabic semantic and syntactic summarisation. In our work we will look at language-dependent tools and apply them to Arabic and English summarisers.

2.1.2 Machine Learning-based Summarisation

Summarisation approaches based on machine learning have been used for classification [Amini and Usunier, 2007; Jaqua et al., 2004], supervised sentence ranking [Fisher and Roark, 2007] and text mining summarisation [Sureka and Kong, 2006]. Figure 2.4 illustrates the methods and techniques used in machine learning. The presented diagram is illustrative; the suggested distribution is not strict, due to the overlap between machine learning approaches.

Machine learning has been used in classification and categorisation of sentences prior to the summarisation process. Jaqua et al. [2004] presented the *ExtraNews* system. They applied generation and classification to produce a very short summary from single and multiple news articles. Categorisation was also proposed in the work by Diemert and Vandelle [2009]. They presented an unsupervised learning technique and cross-reference concept graphs to summarise a massive amount of knowledge derived from unstructured data (e.g., query logs and Web documents).

Support Vector Machine (SVM) is a popular supervised learning method that analyses data and recognises patterns [Cortes and Vapnik, 1995]. The method was applied on automatic summarisation to rank sentences. Li et al. [2007] proposed a multi-document

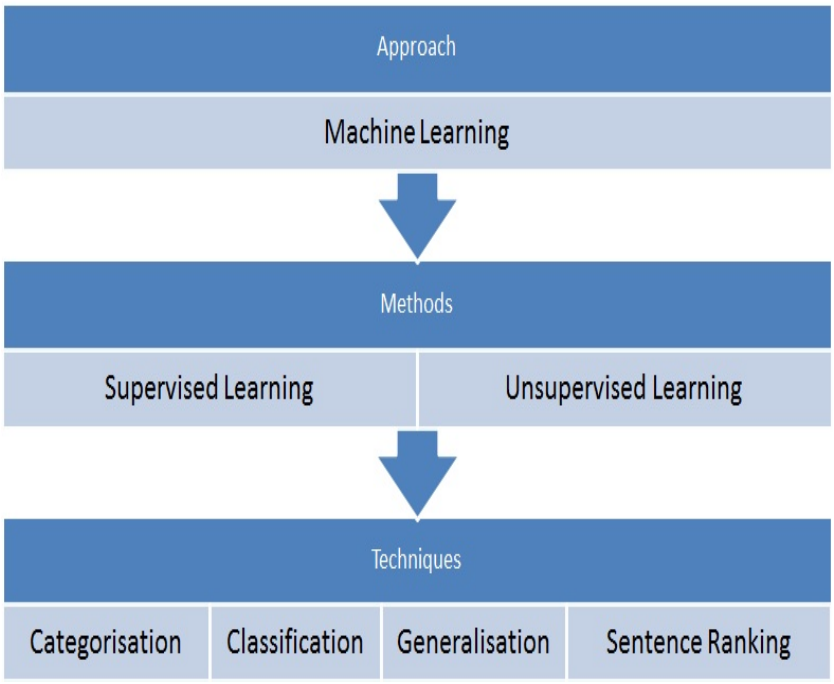

Figure 2.4: Machine Learning Techniques

summarisation systems based on an SVM model. The model was used to automatically combine features and scores of sentences prior to summarisation. Similarly, Galanis and Malakasiotis [2008] used SVM to rank the summary's candidate sentences. Schilder et al. [2008] introduced *FastSum* a query-based summariser based on word-frequency features of clusters, documents and topics. As a query-based summariser *FastSum* ranked the extracted sentences using regression Support Vectors Machine.

Supervised domain-independent ranking was used in the work by Fisher and Roark [2006]. They presented a supervised sentence ranking approach for extractive summarisation. The supervised approach achieved domain independence by making use of a range of word distribution statistics as features.

Sureka and Kong [2006] used machine learning to train their summarisation engine on a financial domain. They implemented a light-weight text mining and a trainable user-focused summarisation engine. Similarly, Amini and Usunier [2007]; Fisher and Roark [2007]; Hickl et al. [2007] applied machine learning on their extractive summaris-

ers. Svore [2007] proposed a single-document summariser that used neural networks for the summarisation process where they extracted a set of features from sentences in a document to identify the importance of the extracted sentence. Ten features have been extracted for each sentence, each feature is chosen to identify characteristics of an article sentence.

Using machine learning in Arabic automatic text summarisation is now starting to attract more attention. Boudabous et al. [2010] presented an automatic summarisation method for Arabic documents. The method was based on applying a numerical approach that used a semi–supervised learning technique. They used SVM for the learning process. They performed a comparative study, using human experts, to evaluate their summariser, which made it hard for us (as we use a standard automatic evaluation metric) to compare our work with them.

Concluding Remarks

Machine learning plays a big role in automatic text summarisation through applying supervised and unsupervised learning methods. Neural networks and SVM are among the methods that have been used to enhance the summarisation quality by suggesting better ranking techniques. Classification and categorisation of sentences prior to the summarisation process have been enhanced by applying machine learning supervised and unsupervised techniques. Machine learning started to attract researchers on Arabic summarisation.

2.1.3 Statistical–based Summarisation

Many summarisation systems rely on statistical methods to extract relevant sentences [Berger and Mittal, 2000; Galanis and Malakasiotis, 2008; Knight and Marcu, 2000]. Figure 2.5 illustrates these statistical techniques, which include Hidden Markov Models,

Katz's K-mixture Model [Katz, 1996], Expectation Maximisation Knight and Marcu [2000] and Vector Space Model [Salton et al., 1975]. The distribution showed in the diagram is illustrative and not strict, due to the overlap between the statistical-based approaches.

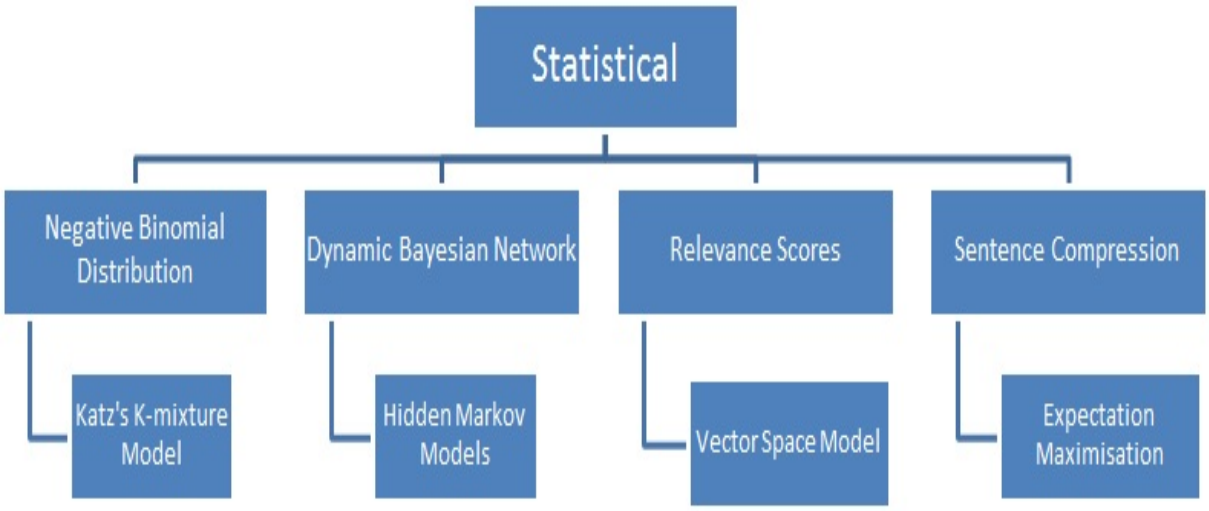

Figure 2.5: Statistical Techniques

Relevance scores, based on a calculation of how important the concept behind the term is to the document, are among the main factors when working with statistical-based summarisation. Alguliev and Aliguliyev [2005] presented a text summarisation method, which created a text summary by defining the relevance score of each sentence and extracted sentences from the original documents. The relevance score of a sentence was determined through its comparison with all the other sentences in the document and with the document title using the cosine similarity measure. The relevance scores are ranked starting with the highest score. Sentences for which the relevance score is higher than a certain threshold value are included in the summary.

Statistical models were also used to enhance the selection or elimination of sentences prior to the summarisation process. Schlesinger et al. [2008] developed a system called *CLASSY* (Clustering, Linguistics, and Statistics for Summarisation Yield). The

system processed each document by applying word and phrase elimination techniques based on a Hidden Markov Model (HMM), which can be considered as a simple dynamic Bayesian network. The model was also used for the sentence selection process to generate a multi-document summary. The HMM used, contained two states, corresponding to *summary* and *non–summary* sentences. They used a naive Bayesian approach to test the probability of a sentence to be in the summary or not. Saravanan et al. [2005] used a negative binomial distribution model known as Katz's K-mixture [Katz, 1996] for term distribution. The model ranked sentences by a modified term weight assignment. Seki et al. [2005] proposed an approach based on sentence extraction, weighted by sentence type annotation and combined with polarity term frequencies. They selected 10 topics related to subjectivity with analysis of "narratives".

The $tf.idf$ (Term Frequency Inverse Document Frequency) [Salton and McGill, 1986] is a simple numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. $tf.idf$ was used in many summarisation tasks as a weighting scheme [Hajime and Manabu, 2000; Wolf et al., 2004]. The $tf.idf$ was adopted by the Vector Space Model (VSM), which is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms [Salton et al., 1975]. Gotti et al. [2007] introduced *GOFAISUM*, a topic-answering and summarisation system. They processed a set of news articles related to a certain topic to produce a 250 word-or-less summary. They used $tf.idf$ scores to find sentences relevant to the certain topic.

Statistical models have also been used to shorten a set of sentences through a process called sentences compression. Knight and Marcu [2000], used the Expectation Maximisation (EM) algorithm to compress sentences for an abstractive text summarisation system. EM is an iterative method for finding Maximum Likelihood (ML) or Maximum A Posteriori (MAP) estimates of parameters in statistical models. In their

summariser, EM was used in the sentences compression process to shorten many sentences into one by compressing a syntactic parse tree of a sentence in order to produce a shorter but maximally grammatical version. Similarly, Madnani et al. [2007] performed multi-document summarisation by generating compressed versions of source sentences as summary candidates and used weighted features of these candidates to construct summaries.

Statistical models were used to maximise total score within a summery length limit. Morita et al. [2011] introduced what they called "query-snowball", a method for query-oriented extractive multi-document summarisation. They worked on closing the gap between the query and the relevant sentences. They formulated the summarisation problem based on word pairs as a maximum cover problem with Knapsack Constraints (MCKP), which is an optimisation problem that maximises the total score of words covered by a summary within a certain length limit.

Concluding Remarks

Relevance score, $tf.idf$, HMM and Expectation-Maximisation were used in many applications such as automatic text summarisation to enhance the selection of important sentences or the elimination of any redundant sentences, which led to more readable summaries and thus summaries better in quality. In part of our work we will also look into vector space model and $tf.idf$ for sentences similarity scores. We also use and apply other statistical similarity metrics such as the Dice's coefficient score [Dice, 1945]. Chapter 5 will show in detail the use of statistical-based models in our single and multi-document summarisation techniques.

2.1.4 Cluster–based Summarisation

Data clustering is the assignment of a set of observations into subsets, so called clusters. Clustering has received a lot of attention in the past years for improving Information Retrieval (IR) [Baeza-Yates and Ribeiro-Neto, 2011; Manning et al., 2008] and to enhance the quality of multi-document summaries, e.g., [Dunlavy et al., 2007]. Clustering has been applied to documents, sentences and words. As shown in Figure 2.6, clustering can broadly be grouped into partitional clustering and connectivity-based clustering[Everitt et al., 2005].

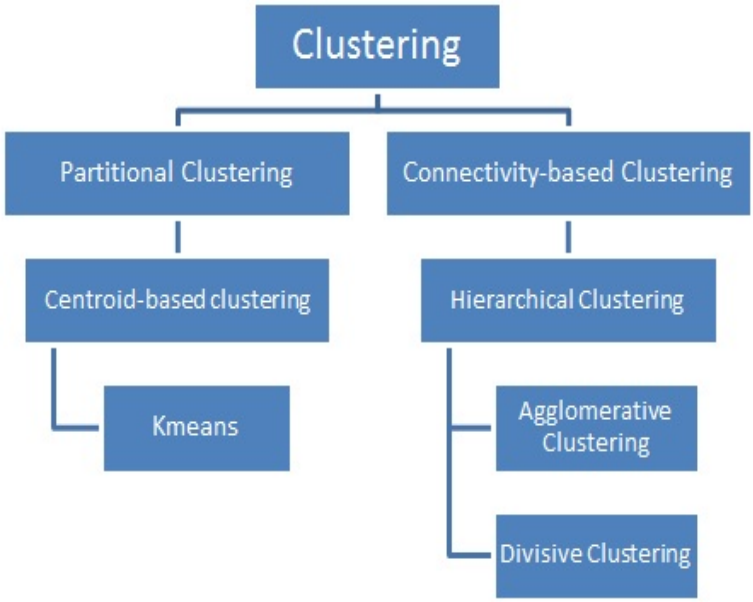

Figure 2.6: Clustering Techniques

In centroid-based multi-document summarisation, which is a form of partitional clustering, similarity to the cluster centroid and the top ranked sentences (ranking differs based on the measure used, e.g., $tf.idf$) has been the main factor in clustering sentences. The centroid is defined as a pseudo-document consisting of words with $tf.idf$ scores greater than a predefined threshold [Radev et al., 2000, 2004].

Kruengkrai and Jaruskulchai [2003] presented a practical approach for extracting the most relevant sentences from the original document to form a summary. Their

approach exploited both the local and global properties of sentences. The local properties can be considered as clusters of significant words within each sentence, while the global properties can be thought of as the relations between sentences in the document. Guo and Stylios [2004] developed a multi-document summarisation system that applied a hybrid of a number of clustering and reduction techniques. They also applied a syntactic parsing method based on link grammar and English syntax [Sleator and Temperley, 1991]. The system was designed to produce a generic summary by not using a user query. It was targeted towards some topics of special interest or purpose no matter whether or not they were closely related to each other. Liu and Lindroos [2006] proposed a Chinese multi-document summariser which was based on clustering paragraphs of the input articles. The number of clusters used changes, based on the number of extracted sentences.

As shown in Figure 2.6 we can see the overlap between the proposed clustering techniques. A combination of these techniques have been used in automatic text summarisation to enhance the quality of the generated summaries. Aliguliyev [2006] proposed a generic summarisation method to extract the most relevant sentences from a source document to create a summary. This method was based on clustering of sentences using language dependent techniques. Wan and Yang [2008] implemented a multi-document summarisation technique using cluster-based link analysis. They used three clustering detection algorithms including k-means, agglomerative and divisive clustering [Everitt et al., 2005]. They clustered sentences, using the three clustering algorithms, into different themes (subtopics), the number of clusters was defined by taking the absolute square root of the number of all sentences in the document set.

Applying hierarchical clustering, Wu et al. [2006] presented a lightweight and rule-free summarisation technique using concept clustering, where sentences describing the same meaning are clustered together. They called it rule-free as they did not build any external knowledge or pre-defined rules. Their method used a two-pass re-ranking

framework. The first pass ordered the concepts which were clustered via conventional top-down clustering algorithm. The second pass generated the representative sentences from the top 10 concepts.

Partitional clustering are commonly used when clustering sentences for automatic text summarisation. Kolla et al. [2007] introduced a multi-document extractive summariser to extract sentences from a given cluster to create summaries. They produced summaries of up to 100 words. To enhance the quality of the generated summaries, they used a partitional clustering technique called Cluster Based Smoothing Model (CBDM), a model that has been used in a cluster-based retrieval process [Liu and Croft, 2004]. Sarkar [2009] presented a sentence clustering based multi-document summarisation system which adopted the incremental clustering method, as used for web clustering [Hammouda and Kamel, 2004]. They reordered the clusters based on their sizes (measured in terms of sentence-counts) assuming the more sentences in a cluster the more important the cluster is.

Although cluster-based methods have shown to be popular in text summarisation, to the best of our knowledge, little work has been reported on applying clustering for Arabic multi-document summarisation. One of a small number of approaches was Schlesinger et al. [2008] who presented a multi-document summariser system. That used K-means clustering algorithm, in addition to other statistical models, to generate multi-document summaries for both English and Arabic languages.

Concluding Remarks

Clustering in automatic text summarisation can be important for both selecting and extracting relevant sentences and eliminating redundancies. In our work we focus on partitional clustering summarisation techniques, or to be more precise, centroid-based clustering as it has been proven to work well with cluster-based summarisation [Everitt

et al., 2005; Kolla et al., 2007; Liu and Croft, 2004; Radev et al., 2000, 2004]. We look into clustering sentences based on their closeness to the centroid. Details on using clustering in our work will be given in Chapter 5.

2.1.5 Other Approaches to Summarisation

A range of other methods have been shown to be helpful for single-document and multi-document summarisation. Automatic summarisation can benefit from work of other tasks such as information retrieval [Kan, 2003].

Mallett et al. [2004] provided an efficient Information Retrieval (IR) oriented technique to extract sentences from text for summarisation purposes. The method extracted key sentences on the premise that the relevance of a sentence is proportional to its similarity to the whole document. Ying et al. [2007] presented a resource and training data-free summarisation model for the DUC multi-document summarisation task. The method simplified the two-pass retrieval [Lemire, 2009] as a passage retrieval task. At first a top-down clustering algorithm was used to merge similar passages, according to the clustering algorithm, into a set of groups. Then the passage retriever extracted relevant groups in response to a given query, where they finally re-form the sentences into the final summary. The passage retriever they used segments each retrieved document into passages and retains the paragraphs that contain at least one of the query terms. Other work involving information retrieval was proposed by[Aker et al., 2010]. In their work they presented A^* search algorithm to find the best extractive summary up to a given length. They also proposed a discriminative training algorithm, which was used to maximise the quality of the best summary. They used a query-based multi-document extractive summariser. The summariser uses a set of different features including query similarity, centroid similarity, sentence position and other features as reported by Brandow et al. [1995]; Conroy et al. [2005]; Edmundson

[1969]; Li et al. [2008]; Radev et al. [2000].

The retrieved information by an IR system can be summarised using different techniques, e.g., sentence compression and clause segmentation. Rasheed et al. [2006] answered user queries by summarising the received information where they compact and reduced sized-information as much as possible. They used aggregate information to answer user's queries, the history of context information is aggregated to generate compact and consolidated context. Others used some similarity metrics such as query expansion to choose an appropriate summary [Nastase et al., 2008; Stokes et al., 2007]. On the other hand, Wan and Paris [2008] proposed a clause segmentation technique to produce summaries. They showed that applying heuristic clause segmentation before sentence selection would improve the summarisation results by reducing the need for sentence compression approaches.

Other techniques have been used in document summarisation, Zajic and Lin [2006] introduced an approach called *Trimmer*, which is a single-document sentence trimming approach. Trimmer was designed with the intention of compressing a lead sentence into a space consisting of tens of characters. Multiple trimmed candidates were generated for each sentence. Sentence selection was used to determine which trimmed candidates provide the best combination of topic coverage and brevity.

The use of off-the-shelf tools is common in document summarisation. Grewal et al. [2003] examined the usefulness of common off-the-shelf compression software, such as *GZIP*, to produce summaries or even enhance already existing summaries. The *GZIP* algorithm works by removing repetitive data from a file in order to compress it, which will benefit in determining which sentences in a summary contain the least repetitive data. To generate the summary they picked the sentence that increased the size of the summary the most, this helped the summary to gain the sentence with the most new information. Liu and Lindroos [2006] presented a process model of topic guided text summarisation–TIDE. By directing the content selection process with identified

topic structure of the text to capture logical relation between the selected sentences. Jin et al. [2008] proposed an approach for summarisation based on content filtering by presenting the definition of dynamic summarisation according to temporal analysis. They then proposed a content filtering model for identification of dynamic information. One of the popular off-the-shelf open source summariser is the Open Text Summariser[1] (OTS). OTS adopts NLP techniques using an English language lexicon with synonyms. It provides rules for stemming and parsing. The NLP tools are used in combination with statistical approaches (e.g., word frequency) for sentence scoring [Boydell and Smyth, 2007]. OTS supports more than 25 languages but to the time of writing and to the best of our knowledge, OTS does not support Arabic. The lack of Arabic NLP tools and resources made Arabic summarisation using OTS unsatisfactory. This is due to the fact that OTS incorporates NLP techniques using a language lexicon with synonyms and cue terms as well as rules for stemming and parsing [Boydell and Smyth, 2007].

Summarised documents could be in any form of text, such as news stories and Internet web pages. Allan et al. [2001] produced a temporal summarising system. They discussed a technology to help a person monitor changes in news coverage over time. They defined temporal summaries of news stories by extracting a single sentence from each event within a news topic. Stories were presented one at a time and sentences from a story must be ranked before the next story can be considered to ensure coherency. Web page summarisation was introduced by [Buyukkokten et al., 2001], the difference is, that they presented a text summarisation for web browsing on handheld devices. They introduced five methods for summarising parts of web pages on handheld devices, such as Personal Digital Assistants (PDAs) and cellular phones. Each web page was broken into text units that can be hidden, partially displayed, made fully visible, or summarised.

[1] `http://libots.sourceforge.net/`

Does a small piece of information work as a summary? One of the state-of-the-art summarisation-like techniques is called "Snippet Extraction", which is usually used by online search engines. Snippets are page excerpts provided together with user query results by search engines as a text summarisation technique [Armano et al., 2012]. Snippet extraction depends on the adopted search engine. Bing[1] search engine provides a tool to extract snippets. When Bing answers a user query, the snippets are displayed in a box when the user hovers the mouse over one of the retrieved results. Yahoo![2] and Bing now work much the same as Bing actually powers Yahoo! search. To the best of our knowledge, Bing developers did not provide information on how the snippets extraction process was performed. Google[3] search engine provides a two-line description snippet, which is usually taken from the description meta-tag [Xiang and Fesenmaier, 2005].

Concluding Remarks

By looking on the related work we can see that automatic text summarisation varies from summarisers for handheld devices to news and web articles. We also see the correlation between the different tools and methods to enhance the sentence retrieval and extraction process, all this suggests that text summarisation is often part of a broader picture. The use of information retrieval in automatic text summarisation could improve the quality of the summarisation techniques by providing ways of enhancing the extraction process. Information retrieval is part of the summarisation system, except that summarisers tends to retrieve sentences rather than documents. Open source summarisers can be helpful to text summarisers as they can be used as a benchmark for other text summarisers or for human summaries. State-of-the-art summarisers showed that recently there has been a renewed interest in automatic summarisation techniques.

[1] `http:\www.bing.com`
[2] `http:\www.yahoo.com`
[3] `http:\www.google.com`

2.2 Evaluation of Summarisation Systems

Evaluating the quality and consistency of a generated summary has proven to be a difficult problem [Fiszman et al., 2009]. One reason is that, in general, there is no obvious ideal summary. The use of system evaluation may help in solving this problem. Two classes of metrics have been developed: form metrics and content metrics. Form metrics focus on grammaticality, overall text coherence, and organisation. They are usually measured on a point scale [Brandow et al., 1995]. Content metrics are more difficult to measure. Typically, system output is compared sentence by sentence or unit by unit to one or more human generated ideal summaries. As with the evaluation of information retrieval [Croft et al., 2009], the percentage of information presented in the system's summary (precision) and the percentage of important information omitted from the summary (recall) are recorded. Automatic evaluation metrics such as ROUGE [Lin, 2004] and BLEU [Papineni et al., 2002] have been shown to correlate well with human evaluations for content match in text summarisation and machine translation. Other commonly used evaluations include measuring information by testing reader's understanding of automatically generated summaries. Human-performed evaluation provides better results, in the sense of accuracy, than automatic evaluation methods, but on the other hand the cost is high [Hirao et al., 2007]. Donaway et al. [2000] proposed sentence-rank-based and content-based measures, which is a measure that associates probabilities with concepts, for evaluating extracted summaries. They compared these techniques with recall-based evaluation measures. They found that content-based measures increase the correlation of ranking. Saravanan et al. [2005] evaluated their multi-document summariser against the frequently occurring sentences in the summaries generated by a set of human subjects. Nomoto and Matsumoto [2001] defined a summary as a set of sentences extracted verbatim from a text, which covered major substance of that text. They took an information-centric approach for evalua-

tion. They evaluated summaries neither in terms of how well they match human-made extracts, nor in terms of how much time does it take for humans to make relevance judgements on them, but in terms of how well they represent source documents in usual information retrieval tasks such as document retrieval and text categorisation. Angheluta et al. [2004] replaced manual evaluation with ROUGE evaluation metric. In their work ROUGE was used to evaluate very short single-document summaries generated by their summarisation system *K. U. Leuven*.

Gold-standard summaries are important for automatic evaluation. Evaluation metrics use gold-standard summaries (usually human summaries) to judge the quality of a system generated summary. The availability of those summaries varies, based on the language and the purpose. In some cases, and especially in domain specific tasks, an evaluation can not be completed without having gold-standard summaries from the same domain (i.e. medical or finance domains). Liu and Lindroos [2006] introduced a series of experiments carried out by adopting varying summarisation schemes for summarising the International Monetary Fund (IMF) staff reports. The summaries produced by the system were evaluated by comparing them to the staff-written executive summaries included in the original reports. Turchi et al. [2010] presented a method for evaluating multilingual multi-document summarisation using a parallel corpus of seven languages. They manually selected the most important sentences in a document cluster in one language then projected the selection to the other languages in the parallel corpus. The method they performed in creating gold-standard summaries was meant for evaluating English extractive multi-document summaries. As mentioned previously, using annotators to manually select sentences could have a high cost and it takes time. The parallel corpora solved this problem so that a gold standard in one language can be projected to other languages, which saved time and minimised annotation effort.

In general, automatic evaluation metrics compare generated summary to a set of

manually created "gold-standards" human summaries. The output of the evaluation process will be a measure of how close the generated system is to the gold-standard summary. At the time of writing, the evaluation metric used for both single and multi-document summaries in the leading conferences is ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [Lin, 2004] and AutoSummENG (AUTOmatic SUMMary Evaluation based on N-gram Graphs) [Giannakopoulos et al., 2008].

ROUGE is based on the general idea of using unigram co-occurrences between summary pairs. This has been shown to correlate well with human evaluations [Lin, 2004]. However, using ROUGE makes it easy to generate summaries that score highly when evaluated. This is because ROUGE is a relatively unsophisticated measure [Sjöbergh, 2007].

ROUGE (ROUGE-N) is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed in Equation 2.1.

$$ROUGE - N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{2.1}$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. ROUGE is a recall-based measure. As in Equation 2.1, the denominator of the equation is the total sum of the number of n-grams occurring at the reference summary side [Lin, 2004].

AutoSummENG is based on comparing the character n-gram graphs representation of extracted summaries and a number of model summaries. It calculates a *CharGraphValue*, which indicates how much the graph representation of a model summary overlaps with a given peer summary, taking into account how many times two N-grams are found to be neighbours.

The automatic summarisation workshops and conferences[1] have adopted ROUGE and AutoSummENG-MeMoG (first used in 2011) [Giannakopoulos and Karkaletsis, 2011]. The reason behind using ROUGE as a standard metric in most of the summarisation conferences is because it correlates well with human evaluations for content match in text summarisation. ROUGE was used to evaluate extractive and abstractive summaries, but it showed to correlate better for extractive ones [Liu and Liu, 2008]. At the time of writing, a number of evaluation metrics and methods have been used in one of the latest workshops on text summarisation, the *MultiLing Summarisatin Pilot 2011*[2]. The workshop task was a multi-document, multi-lingual summarisation pilot. The evaluation of results was performed both automatically and manually. The manual evaluation was based on the *Overall Responsiveness* of a text and the automatic evaluation used the *ROUGE* and *AutoSummENG-MeMoG* methods to provide a grading of performance. For the manual evaluation the human evaluators were provided with the following guidelines: Each summary is to be assigned an integer grade from 1 to 5, related to the overall responsiveness of the summary. A text should be considered to be worth a 5, if it appears to cover all the important aspects of the corresponding document set using fluent, readable language. A text should be assigned a 1, if it is either unreadable, nonsensical, or contains only trivial information from the document set. The content and the quality of the language are considered equally important in the grading. Systems participating at the workshop should generate summaries of size between 240 and 250 words inclusive. Summaries that are out-of-limit are penalised using the Length-Aware Grading measure (LAG). Given a summary S of length $|S|$ (in words) assigned a grade g, a lower word limit count l_{min} and an upper word limit

[1] http://www.nist.gov/tac/

[2] http://www.nist.gov/tac/2011/Summarization/index.html

count l_{max}, then LAG is defined as in Equation 2.2.

$$LAG(g, S) = g * \big(1 - \frac{\max(\max(l_{min} - |S|, |S| - l_{max}), 0)}{l_{min}}\big) \tag{2.2}$$

In the task specific evaluation, $l_{min} = 240, l_{max} = 250$. LAG simply provides a linearly diminishing weight to grades diverging from the limits. For extreme text sizes ($|S| > l_{min} + l_{max}$), LAG may even have a negative value. Such a case never appeared in the MultiLing pilot. The LAG function was applied to the Overall Responsiveness score in the analysis of performance, therefore LAG in the following sections implies LAG of the Overall Responsiveness. The automatic evaluation at the workshop was based on human-generated model summaries provided by fluent speakers of each corresponding language (native speakers in the general case). ROUGE variations (ROUGE1, ROUGE2, ROUGE-SU4) and the MeMoG variation of AutoSummENG were the main automatic evaluation metrics used in the MultiLing workshop.

Concluding Remarks

Automatic evaluation metrics have been used widely to replace and reduce the cost and effort required when performing human evaluation. Most of the automatic evaluation metrics such as ROUGE and AutoSummENG have been proven to correlate well with human evaluations. ROUGE is considered the standard evaluation metric for automatic text summarisation and has been used as the main evaluation metric in many of the leading conferences on automatic text summarisation. In our work we use different evaluation metrics including ROUGE and AutoSummENG and the MeMoG variation. We participated in the MultiLing Summarisation Pilot and evaluated our summarisers using the above mentioned automatic evaluation metrics. In addition to those metrics we also use Dice's coefficient as a similarity measure. We evaluate our summarisers using human experts, by having native Arabic speakers and native English speakers to

evaluate our Arabic and English summarisers.

2.3 Arabic Natural Language Processing (ANLP)

2.3.1 The Arabic Language

The Arabic language is the largest living member of the family of Semitic languages in terms of speakers. It is closely related to Amharic and Aramaic. Arabic is spoken by around 400 million people living in the Middle East, North Africa, and the Horn of Africa [Prochazka, 2006]. With this huge number of Arabic speakers the focus toward processing this language is increasing rapidly. Like other languages, literary Arabic continues to evolve. Classical Arabic (especially from the pre-Islamic to the Abbasid period, including Qur'anic Arabic) can be distinguished from Modern Standard Arabic (MSA) that is used nowadays in news and media. Arabic has several dialects (varieties of Arabic) mostly spoken and rarely written in contrast to the MSA which is mostly written and rarely spoken.

A few researchers have tackled the problem of Arabic syntax with some success [Al-Shammari and Lin, 2008; Benmamoun, 2007; Sawalha and Atwell, 2010b]. Yaseen and Theophilopoulos [2001] proposed NAPLUS, which is a prototype system for processing and understanding the Arabic language. They developed utilities and tools to support Arabic NLP research in many areas such as automatic translation, text abstraction, question answering interfaces to databases and many related applications. Researchers show a growing interest in working on Arabic language processing for a number of reasons. Although dialects differ from one country to another and even in the same country, there is only one written language. The number of Internet users in the Arab world increased in 2011 to reach 77 millions[1]. In the last decade, the volume of Arabic

[1]http://www.internetworldstats.com/

textual data has also grown on the Internet and Arabic software for browsing the Web has improved.

2.3.2 Challenges in Arabic NLP

Semantic processing for Arabic language tends to be more complex than it is for English and other European languages. The Arabic language is highly derivational; even with an annotated corpus it is still complex to rely on a "bag of senses" approach to determine relevancy. Synonyms in Arabic are treated differently. A word can hold up to seven synonyms, but the replacement of any of these words depends on the context, where this could lead to extracting irrelevant sentences. In the Arabic language there is a property called "borrowing" where a certain word can be borrowed to be used in a different context. Borrowing in Arabic is similar to metaphor in English but it has nothing to do with idioms. Borrowing is commonly used in written text and day to day conversations. It is complex for a machine to sense the meaning. Highly complex morphology and dialectal differences are some of the reasons for the lack of Arabic semantic processing tools. Many of the Arabic resources and tools available online are commercial, example on those are Sakhr[1] and RDI[2]. There has been an increased demand for resources and tools to assist and advance the research on Arabic. Although there are some current annotated Arabic text treebanks (e.g. Arabic Penn Treebank[3] and Prague Arabic Dependency Treebank[4]), these were not used in the most advanced summarisation conferences (e.g. Text Analysis Conference[5] and Document Understanding Conference[6]) as they lack any Arabic gold standard summaries.

There are some aspects that slowed down the progress in Arabic NLP compared

[1]http://www.sakhr.com/
[2]http://www.rdi-eg.com/technologies/arabic_nlp.htm
[3]http://www.ircs.upenn.edu/arabic/
[4]http://ufal.mff.cuni.cz/padt/PADT_1.0/
[5]http://www.nist.gov/tac/
[6]http://duc.nist.gov/

to the accomplishments in English and other European languages [Diab et al., 2007; Roberts et al., 2006], which include the following.

- The absence of capitalisation in Arabic, makes it hard to identify proper nouns, titles, acronyms, and abbreviations.
- Arabic is highly inflectional and derivational, which makes morphological analysis a very complex task.
- Diacritics (vowels) are, most of the time, omitted from the Arabic text, which makes it hard to infer the word's meaning and therefore, it requires complex morphological rules to tokenise and parse the text.

  In addition to the above linguistic issues, there is also a shortage of Arabic corpora, lexicons and machine-readable dictionaries. These tools are essential to advance research in different areas.

Arabic NLP has focused on the manipulation and processing of the structure of the language at morphological, lexical and syntactical linguistic levels. Semantic processing of the Arabic language is still in its early stages [Al-Shammari and Lin, 2008; Diab et al., 2007; Haddad and Yaseen, 2005; Hmeidi et al., 2010]. Morphological analysis of words in a text is the first stage of most natural language applications. It has been regarded as being particularly crucial in processing highly inflectional and derivational languages like Arabic [Diehl et al., 2012]. Although Arabic morphology is considered complex, researchers on ANLP were successful in tackling this complexity problem [Abuleil et al., 2002; Beesley, 1998; Darwish et al., 2005; Habash and Roth, 2011; Sawalha and Atwell, 2010a; Smrž, 2007]. Arabic morphological analysis, like other languages, involves tokenisation and stemming, which have been investigated and tools were created by many researchers on ANLP [Al-Ameed et al., 2006; Al-Shammari and Lin, 2008; Attia, 2007; Hmeidi et al., 2010; Khoja and Garside, 1999; Larkey et al., 2002].

2.3.3 Arabic NLP Tools

In any natural language processing applications tokenisation is required before any morphological analysis or parsing can proceed. A tokeniser takes input text and divides it into "tokens" or separate words. The tokenisation uses units (delimiters) to divide the text, such as words punctuation, numbers, dates, etc.

Stemming determines the morphological stem of a given inflected word. It uses morphological rules or heuristics to remove affixes from words before indexing. Stemming reduces the number of indexing terms and reduces redundancy (by collapsing together words with related meanings). Stemmers are common elements in information retrieval systems, since a user who runs a query on "computers" might be also interested in documents that contain the word "computer" or "computerisation" [Croft et al., 2009]. In Arabic, stemming is more difficult as most of the Arabic words are derived from a few thousand trilateral roots [Wightwick and Gaafar, 1998]. Prefixes, infixes and suffixes can then be added to the word to indicate its number, gender and tense. Arabic stemming is the process of removing all the affixes (prefixes, infixes and suffixes) from a word to extract its root. On the other hand, Arabic light stemming is the process of removing a small set of prefixes and suffixes with no attempts to remove infixes or returning the word's root [Al-Ameed et al., 2006]. An Arabic stemmer for example should identify the string, *dAres* "دَارِس" (studier), *dars* "دَرس" (lesson), *madrasaT* "مَدرَسَة" (school), *mudarres* "مُدَرِّس" (teacher), as based on the root *drs* "دَرَس" (he studied).

2.3.4 Arabic Summarisation

Summarisation research on Arabic documents has not advanced as fast as work on other languages such as English. Douzidia and Lapalme [2004] drew attention to Arabic summarisation when they developed a system that summarises Arabic documents

by building an abstract representation of the whole document, and then generating a shorter text by selecting a few relevant sentences of the original text. Their System, called "Lakhas", was developed using extraction techniques in order to produce ten word summaries of news articles. Words were being weighted using a similarity coefficient technique. As this work involved working on Arabic we find it related to our research. There are some differences from what we are working on, especially the idea of generating very short summaries. "Lakhas" participated in the DUC-2004[1] summarisation competition to generate very short summaries (75-bytes). In order to achieve that, Douzidia and Lapalme [2004] used four sentence-reduction method to compress sentences and make them shorter. We believe that the reduction process led to the loss of valuable information. DUC-2004 tasks were only applicable to English summarisers, to overcome this problem Douzidia and Lapalme [2004] translated the DUC-2004 English dataset into Arabic using a machine translation tool. To be able to evaluate the summaries, they translated the generated summaries from Arabic back to English. This repetition of the translation process may have resulted in incoherent sentences. The main difference between our work and Lakhas is the process of computing words' weight. Lakhas computes the weights reflecting the importance of a word to the document collection rather than the summarised document. In our work we overcome this problem by computing the weight of the words to reflect their importance to the sentences in the summarised document. The method Lakahs introduced will always result in the same weight for a single word, even when summarising a different document. In our work the weight of a word is dynamic and changes according to the sentence and the document the word appears in, as the similarity coefficient is concerned with the frequency of the words.

Other work on Arabic summarisation include *CLASSY* summariser (see Section 2.1.1), which relied on automatically translating the Arabic dataset into English. The

[1] http://www-nlpir.nist.gov/projects/duc/guidelines/2004.html

translation of Arabic documents performed resulted in inconsistent sentences as core keywords may have been dropped when translating. Errors in tokenisation and incorrect sentence-splitting were among the main challenges.

Generally speaking, we introduce solutions to problems facing Arabic single and multi-document summarisation, by tackling the lack of resources and experiments for those fields. We push forward Arabic multi-document summarisation which has started to attract more attention in the research community. An example for that trend is the inclusion of Arabic as one of the languages in the new TAC MultiLing pilot track[1].

Concluding Remarks

The Arabic language is highly inflectional and derivational, which has led to difficulties in pushing forward the research on Arabic natural language processing. Lack of resources and tools are among the challenges facing the automatic processing of the Arabic language. Arabic automatic summarisation is not so advanced when compared to English and other European language, this is due to the challenges mentioned above. In our work we use different NLP tools for both Arabic and English to advance the state-of-the-art in summarisation. The tools include tokenisers, light and root stemmers in addition to stop-word removal and latent semantic analysis. Chapter 5 shows the description of the summarisers and the experimental setup. The evaluation of the summarisers is given in Chapters 6 and 7. We believe that the results presented in this work can help push forward the research on Arabic text summarisation. To overcome the lack of resources problem, we created Arabic single-document and multi-document gold-standard summaries as one of the contributions of the work described in this research. Details on the creation of the resources are given in Chapter 4.

[1]http://www.nist.gov/tac/2011/Summarization/

2.4 Summary

The approaches illustrated in this chapter showed the variety of techniques and tools applied to text summarisation. The related work showed that many statistical, clustering, semantic, syntactic and machine learning methods have been applied to automatic summarisation. The related work showed a small amount of work done on the Arabic language, especially in the multi-document summarisation field. The chapter pointed out the lack of Arabic resources needed to advance the research in the summarisation field. The chapter concluded by showing the state-of-the-art evaluation metrics for automatic summarisation. State-of-the-art summarisers showed, that recently, there has been a renewed interest in automatic summarisation techniques. Automatic text summarisation overlaps with many other natural language processing fields such as information extraction, automated question answering, natural language generation and machine translation. The overlap suggests that automatic text summarisation is actually a part of a larger picture.

Chapter 3

A Framework for Automatic Summarisation

This chapter presents the general framework of automatic summarisation, which is necessary to understand our contributions in this area. In this chapter we discuss the abstract architecture of an automatic summarisation system. The architecture described in this chapter is general and can be applied to both Arabic and English languages. Summarising Arabic and English test collections is important for this work. We can compare the summarisation techniques we use and see how Arabic summarisers perform when compared to English summarisers. Later in the chapter, we talk about single-document and multi-document summarisation and our contributions to the Arabic text summarisation field, especially to the field of multi-document summarisation. The section shows the novel methodologies and models used to summarise Arabic text and eliminate redundancies. The chapter concludes with a section discussing the use of Natural Language Processing in text summarisation.

3.1 General Architecture

Figure 3.1, illustrates the general architecture of an automatic Arabic text summarisation system. The presented architecture is language-independent and can be applied to other languages. This is because, in our work, we present a number of English summarisers for comparison purposes. The architecture is for both single and multi-document summarisation approaches. The user-interface represents any interaction between a user and a document provider service, which could be a simple concordancer [Roberts et al., 2006] system or a search engine. The user selects either a single document or a group of related documents. The summarisation process produces a shorter version of the selected document(s). The process ends by generating a single summary. The index is a repository, where the document's information is stored and retrieved. Such information could be the sentence's weight, position and extracted words. This is presented back to the user through the user-interface.

The general architecture diagram can be split into two general modules, the first is the *Document Selection Module*, which could be any method concerned with finding a relevant document from a document collection to satisfy a user's need. The second is the *Document Summarisation Module*, which could be in any form. For example, one could consider using a search engine to retrieve documents where a user decides on which document or group of documents to be summarised. We are most interested in the *Document Summarisation Module*, the structure differs based on the purpose and the techniques used. To give an abstract explanation, this module starts when the user selects a document or a group of documents as an answer to the information sought. A common (generic) approach is illustrated in Figure 3.2. Each of the selected documents undergoes a splitting process where the document is being split into sentences based on a certain delimiter. The sentences then go into a sentence matching process, where they are matched against other information, for example, a user query or the first sentence

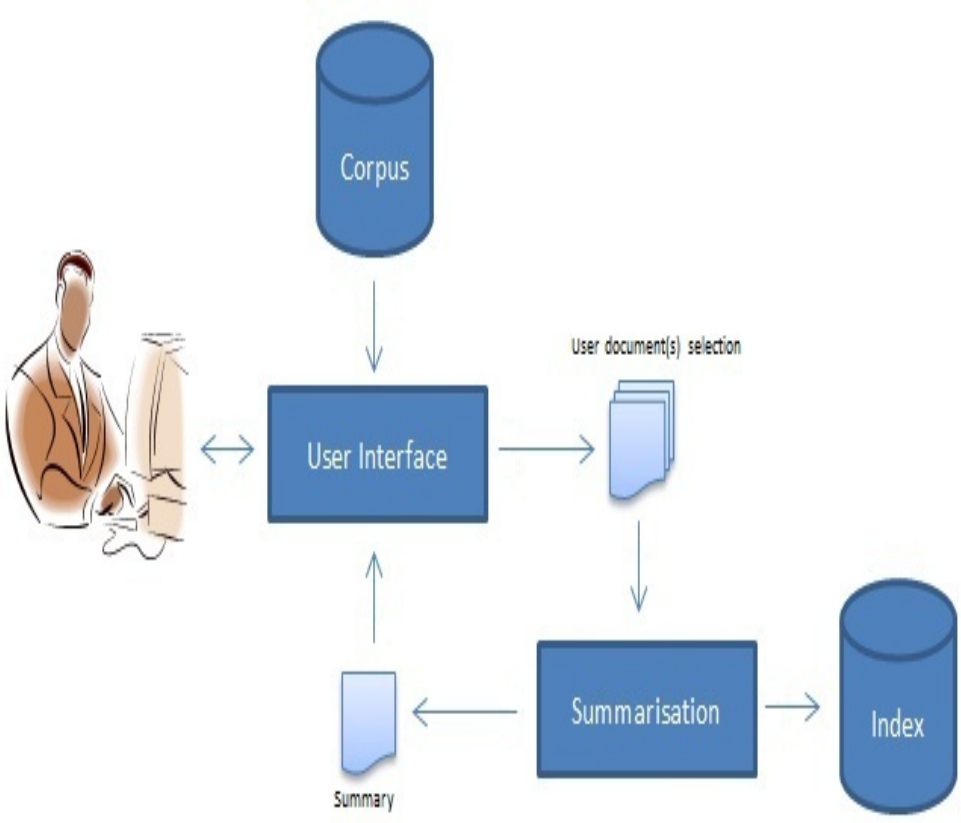

Figure 3.1: Summarisation: General Architecture

in a document. The sentence selector process is based on ranking the sentences after the matching process. Based on a certain threshold and up to a certain limit, the top sentences are selected and sent to the fusion process where the sentences are being ordered (e.g. according to their position in the document) and combined to generate a summary.

The architecture described above is a very abstract illustration of most summariser systems regardless of the language, approaches, tools or techniques used. In this research we are mainly interested in the summarisation process. The *Document Selection Module* (retrieving documents) has been discussed in detail in [Baeza-Yates and Ribeiro-Neto, 2011; Manning et al., 2008]. The summarisation process is preceded by steps that include the retrieving process of the documents and the data-collection pre-processing phase. Below we talk about the general pre-processing tools to process text

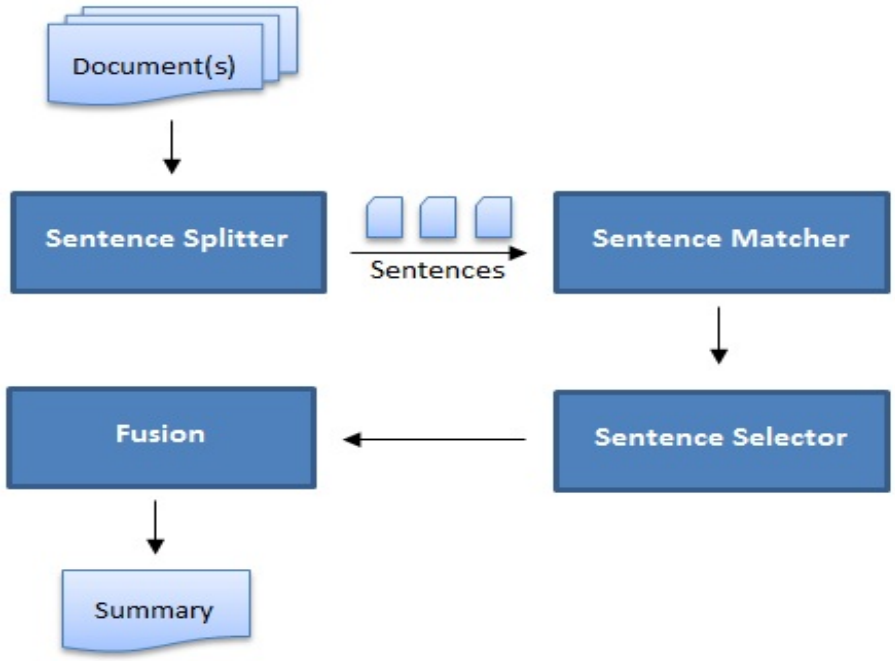

Figure 3.2: Summarisation Selection Module

collections used for automatic summarisation purposes.

3.2 Data Collection Pre-processing

The summarisation process requires a number of phases before generating a summary. These phases include document and language processing. The document, or a group of documents to be summarised, needs to first undergo a number of pre-processing steps. The abstract phases include tokenisation process, indexing the documents, stop-words removal and stemming.

3.2.1 Document Indexing

Indexing has been used for a long time as an efficient means of accessing information (i.e. indices in books, telephone directories, libraries, etc.). Indexing can be done either manually, by selecting index terms from a document by hand, or automatically. Manual

indexing is a time consuming and error prone process. Automatic indexing, therefore, has been widely used in experimental systems such as information retrieval systems and it has been proven to produce better results than manual indexing [Salton, 1989]. In the summarisation process creating a summary is mostly influenced by the techniques used and the creation of a good index of terms. The goal of indexing for information retrieval and automatic summarisation is to find, and select, terms to describe the content of a document as closely as possible. Therefore, the success of selecting the sentences is determined by a careful selection of index terms (e.g., avoid common words or what are called "stop-words") [Allan et al., 2003; Croft et al., 2009; Ganapathi and Zhang, 2011]. The information in the index will be used by the *sentence-matcher* process (see Figure 3.2) to decide on which sentences to include in the summary. Figure 3.3 shows the abstract process of creating an index for document summarisation. As shown in Figure 3.3, the indexing process starts with automatically selecting documents from the document collection set (the selection is done automatically by going through the document set and selecting documents that are not yet indexed). During this process information about the document will be sent to the index, including the document's title, size, location and genre if present. The selected document will be split into sentences using delimiters (e.g. full–stop, question–mark, exclamation–mark). Finally those sentences will be split into tokens based on delimiters (e.g. white space). The tokens will be indexed and information about each token's location, position, frequency and weight will be recorded.

The information in the index will be used in the ranking process where information about each word's weight will be used to calculate the similarity between sentences. In general, *weights* reflect the relative importance (based on factors such as the term-frequency of this token) of the extracted tokens in documents. In the field of automatic summarisation, in some cases, the weight will reflect the relative importance of those tokens in sentences, as the retrieving process is more concerned with returning relevant

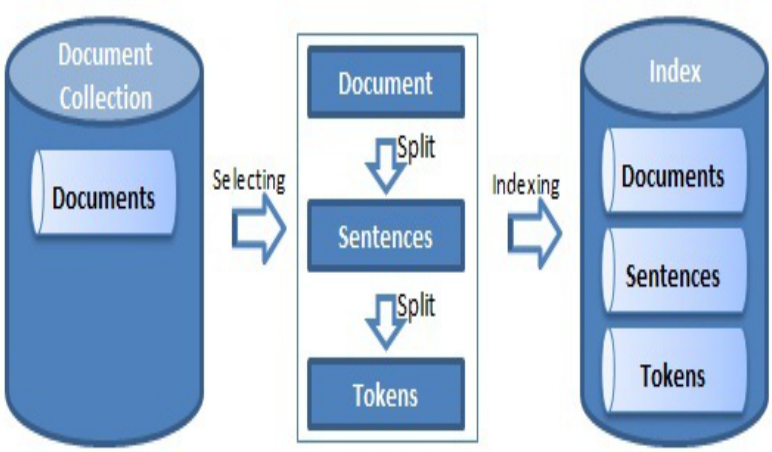

Figure 3.3: Index Creation Process

sentences rather than the documents themselves. One of the most common models to compute terms weight is the $tf.idf$ weighting, which makes use of two factors: the term frequency tf and the inverse document frequency idf [Salton and McGill, 1986]. The weight w of term j in a document i can be computed as:

$$W_{ij} = tf_{ij} * idf_j \tag{3.1}$$

where

$$idf_j = \log \frac{n}{df_j} \tag{3.2}$$

and tf_{ij}, the term frequency, is the number of times that term j occurs in document i, n is the number of documents in the collection, and df_j is the document frequency of term j [Grossman and Frieder, 2004].

Tokenisation

For any input document the pre-processing starts with tokenisation, the process of splitting the document into tokens (Algorithm 1). The results of the tokeniser correspond to units whose character structures are recognisable, such as punctuation, numbers, dates,

etc. Arabic tokenisation may involve using additional Arabic punctuation characters which are not present in the English language as for example the Arabic question mark (؟).

D_i represents the i^{th} document in a Text Collection of n documents;
S_j represents the j^{th} sentence in D_i, m is the number of sentences in D_i and j: 1 to m;

input : Text Collection TC
output: List of all extracted words W_1, W_2, ..., W_h from TC

```
foreach D in TC do
    Begin;
    Split D into sentences;
    foreach S in D do
        Begin;
        Split S into words;
    end
end
```

Algorithm 1: Tokenisation Algorithm

Stemming and Stop-word Removal

Stemming can be applied to reduce the index words to their stems. This is done by automatically stripping affixes from words to obtain stems. For example, several words that are used to express a particular concept (e.g., *works, worked, working, and worker*) can be grouped together and stemmed to *work*, since they all have the same conceptual meaning. As in information retrieval [Croft et al., 2009], stemming could be helpful for automatic summarisation through finding semantically related words, which will help in selecting more sentences. One type of stemming is called "light stemmer" (see Section 2.3.3). The other type is called "root extractor" stemmer, the so called stemmer removes suffixes, infixes and prefixes and uses pattern matching to extract a word's root. The root is usually formed of 3 letters as in Arabic language [Al-Shammari and Lin, 2008; Khoja and Garside, 1999].

Croft et al. [2009] reported that using a stemmer for IR when retrieving English sentences has little improvement on the quality of results, unlike the use of a stemmer for Arabic which showed a better improvement.

In computing, stop-words are words which are filtered out prior to, or after, processing of natural language data (text). These may include prepositions, determiners and common words that appear frequently in any text. Working with extractive summarisation systems we are looking for words that could help in extracting relevant sentences. When creating the index we tried to record important information and ignore less important information to reduce the index size. This can be achieved by skipping prepositions, determiners and common words from the text analysis process. Those words are called "stop-words" as the text processing stops when detecting any of them. There are basically two advantages of ignoring words of this type. First, this process reduces the length of the index by about 40% [Gancarski et al., 2006]. Second, it allows computing similarity between sentences to be more realistic [Salton and McGill, 1986]. Stop-word lists differ based on the language the list is created for. In other cases, and when working on a domain specific summarisers, stop-word lists could contain words that are considered non-stop-words in other domains. There are many ways to create a stop-word list. One common technique is by processing the text collection and recording the frequency of all the terms, and then considering all words with frequency greater than a certain threshold as stop-words [Fox, 1989]. Algorithm 2 shows a general language independent stemming and stop-word removal process. The algorithm is illustrative; the suggested use of words is not strict. Other information can be applied (e.g., noun-phrases, verb-phrases, named-entity definitions, etc).

D_i represents the i^{th} document in a Text Collection of n documents;
S_j represents the j^{th} sentence in D_i, m is the number of sentences in D_i and j: 1 to m;
W_a represents the a^{th} word in S_j, b is the number of words in S_j and a: 1 to b. Words can be replaced with any other information (e.g., noun-phrases, named-entity definitions, etc);
Let G be the minimum-length of a word W_a in a language V;
Let $STOPS$ be the stop-word list for language V;

input : Words extracted from the Tokeniser W_1, W_2, ..., W_h
output: Stem of the input word

```
foreach W ∈ S do
    Begin;
    if W Not In STOPS then
        Index W;
        if Length of W<G then
            Word can not be stemmed;
        else
            Remove prefixes, suffixes and infixes from W;
        end
    else
        Do not index W;
    end
end
```

Algorithm 2: Stop-word Removal and Stemming Algorithm

3.3 Single-Document Summarisation

One of the early approaches towards automatic summarisation is single-document summarisation. In its basic definition, single-document summarisation is the process of creating a summary for a single text document. There is no standard length for the generated summary, it varies based on the implementation guidelines.

3.3.1 General Overview

A number of different techniques have been applied to single-document summarisation. Both language-dependent and language-independent approaches have been proposed.

For a summariser to be language-independent it should be portable to new languages or domains [Mihalcea, 2005]. Language-independent approaches to single-document summarisation do not rely on any language-specific knowledge resources or any manually constructed training data. Language-dependent approaches use language-specific tools (e.g., parser, knowledge-base, lexical chains, etc.) to find semantic similarities between sentences.

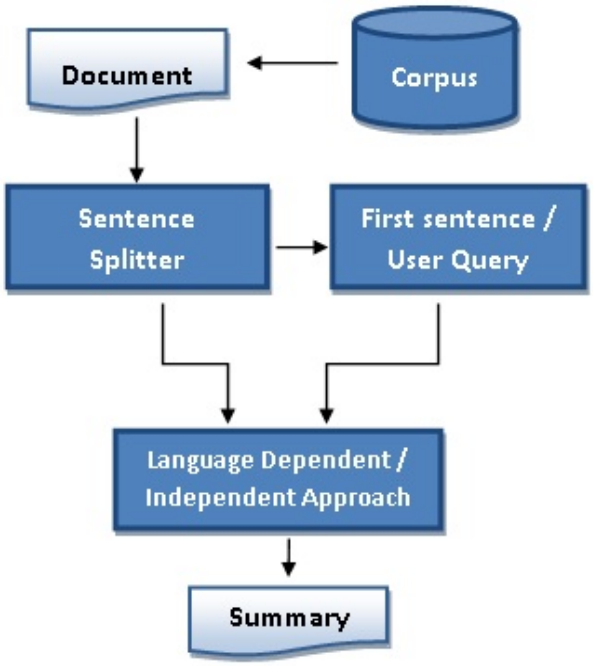

Figure 3.4: Single-document Summarisation Architecture

Single-document summarisers types include *generic* and *query-based* (query-focused) [Zha, 2002]. Generic single-document summarisers match a document's sentences to a certain information extracted from the document itself, this could be the document's title or the first sentence in the document. Generic summarisers are more useful for long documents containing a variety of topics [Zha, 2002]. Query-based single-document summarisers use a user query to summarise the document around this query. Sentences in the document are matched to the user query using similarity measures. Sentences that are closer to the query are being selected to be in the summary.

D_i represents the i^{th} document in a Text Collection of n documents;
S_j represents the j^{th} sentence in D_i and m is the number of sentences in D_i and j: 1 to m;
$SM[i, j]$ represents a two dimensional array, to store similarity values;
$Similarity(S_j, I)$ computes the similarity value between S_j the j^{th} sentence in a document D_i and other information I in this document (e.g., first sentence);
N represents the top maximum number of sentences or words allowed in the summary based on the similarity value;

input : Text Collection TC
output: A Summary for each document in TC

```
foreach D in TC do
    Begin;
    Split D into sentences;
    foreach S in D do
        Begin;
        SM[i,j] = Similarity(S, I(D));
    end
    while Not end of SM[i,j] do
        Begin;
        Read SM[i,j];
        Order SM[i,j] descending;
        Select and combine the top N sentences or words;
        Generate Summary;
    end
end
```

Algorithm 3: Single-document Summarisation: General Algorithm

Figure 3.4 illustrates the general architecture for a single-document summariser. As illustrated, the summarisation process starts by selecting a single document from a text collection. The process of selecting the document varies, it could be a simple concordance system or a sophisticated information retrieval system. The selected document will undergo a splitting process to divide the document into sentences. Depending on the single-document summariser type (generic or query-based), the summariser uses the user query, or for example the document's first sentence in the summarisation process. The sentences in the document are matched to the query or the first sentence using language dependent or language independent tools. The matching process returns the

most important sentences (most similar to the query or the first sentence) to generate the final summary. Algorithm 3 illustrates the basic breakdown of the single-document summarisation process.

3.3.2 Contribution and Methodology

We consider single-document summarisation first, as it forms a backbone for our work on multi-document summarisation. Experimenting with single-document summarisation, we present different methods and techniques to summarise a single document. We introduce generic and query-based single-document summarisation approaches. We investigate two generic summarisers *Gen–Summ* and *LSA–Summ* and two query-based summarisers *AQBTSS* and *ACBTSS*. We apply statistical and semantic techniques and we use basic natural language processing tools. The novelty of this work consists of the systematic process of creating resources, implementing single-document summarisers and evaluating the summaries, using both human and automatic evaluation processes. In the proposed summariser we enhance the techniques previously applied to Arabic single-document summarisation (See Section 2.3.4). In Chapter 5 we show the approaches and the descriptions of the summarisers. Chapter 6 shows the evaluation results of these summariser and the comparisons between the summarisers and the different evaluation metrics. In Section 6.3 we study the effect of the summary length on the evaluation scores. At the time of writing and to the best of our knowledge, no work has been reported addressing the effect of the summary length on the scores by the automatic evaluation metrics. The process of creating single-document summaries corpus is given in detail in Chapter 4. The need for such a corpus emerged because of the lack of any Arabic resources needed to evaluate Arabic single-document summarisers.

3.4 Multi-Document Summarisation

Multi-document summarisation is the process of generating a single summary for a set of related articles. This could include news articles, publications and financial reports. Multi-document summarisation by humans is a time consuming job as it requires reading the complete set of documents and then summing up the key events in a way to ensure coherence and maintain relevant relationships. Automatic extractive multi-document summarisers are capable of swiftly extracting important sentences from a set of related documents and generate a single summary.

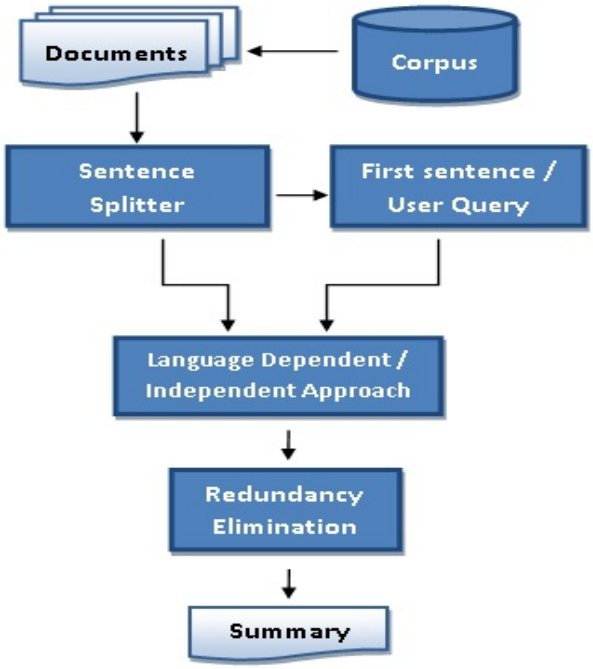

Figure 3.5: Multi-document Summarisation Architecture

3.4.1 General Overview

Figure 3.5 illustrates the abstract overflow diagram of a general multi-document summarisation process. As illustrated in the diagram, language dependent and language independent approaches are applied on the documents after they undergo the sentence

D_i represents the i^{th} document in RA, n is the number of tested documents and i: 1 to n;
S_j represents the j^{th} sentence in D_i and m is the number of sentences in D_i and j: 1 to m;
$SM[a, b]$ is a two dimensional array, to store similarity values;
$Similarity(S_j, I)$ computes the similarity value between S_j the j^{th} sentence in a document D_i and other information I in this document (e.g., first sentence);
U represents the maximum number of sentences to be selected from each document in the set of related articles;
N represents the maximum number of sentences or words allowed in the summary;

input : A set of related articles RA
output: A summary for the set of related articles

```
foreach D_i in RA do
    Begin;
    Split D_i into sentences;
    foreach S_j in D_i do
        Begin;
        SM[i, j] = Similarity(S_j, I);
    end
    while Not end of SM[a, b] do
        Begin;
        Read SM[a, b];
        Order SM[i, j] descending;
        Eliminate Redundancy;
        Select top (U) sentences from each document;
        Select and combine the top N sentences or words;
        Generate Summary;
    end
end
```

Algorithm 4: Multi-document Summarisation: General Algorithm

splitting process, the approaches include tools and techniques that work for both extracting important sentences and eliminating redundant ones. Algorithm 4 illustrates the general phases of generating a summary for a set of related articles.

The same summarisation approaches and types used in single-document summarisation can still be applied in the multi-document summarisation process. However, the work on multi-document summarisation is more complex, this is due to the fact

that we are dealing with more than one related document at the same time. Having to summarise multi-documents may lead to redundant sentences emerging. Having a set of articles discussing the same topic could result in redundant sentences when generating a multi-document summary. In order to come up with a sensible summary we first need to eliminate those redundant sentences. Many tools and techniques have been proposed to eliminate redundant sentences in multi-document summarisation, see Chapter 2. The proposed tools include language dependent approaches, such as deep semantic analysis, and other language independent approaches using statistical models and similarity measures, to define which sentences are redundant and should be eliminated. For example, when comparing two sentences, one of them is considered redundant if the similarity value between these two sentence is high (based on a selected similarity threshold). Only one of the compared sentences should be selected. The decision on which sentence should stay and which one should leave is based on the redundancy elimination technique or tool used.

Working with multi-document summarisation opens questions on how to solve the redundancy problem without eliminating crucial sentences and which order should the extracted sentences follow. Different methods to order the extracted sentences include, chronological order of the events in the extracted sentences, sentence-position in the documents and order sentences according to a similarity measure (the most similar appear on top) [Barzilay et al., 2001].

3.4.2 Contribution and Methodology

Experimenting with multi-document summarisation, we present different methods and techniques to summarise multiple documents. We introduce statistical-based, semantic-based and cluster-based multi-document summarisation approaches. The redundancy elimination method introduced in our work is novel and has not been applied to state-

of-the-art Arabic and English multi-document summarisers. The novelty of this work consists of the systematic process of creating resources, implementing multi-document summarisers and evaluating the summaries using both human and automatic evaluation processes. In Chapter 5 we show the approaches and the descriptions of the summarisers in detail. Chapter 7 illustrates the evaluation results of different multi-document summariser, the comparisons between the summarisers and the different evaluation metrics. We also compare the evaluation results with state-of-the-art Arabic and English summarisers. We created multi-document summaries corpora to address the shortage of Arabic resources needed to advance the work on multi-document summarisation (Chapter 4). At the time of writing, the proposed methods and techniques presented in this work have not been applied on Arabic language for the purpose of multi-document summarisation. We believe that the work presented in this thesis could help in advancing the research on multi-document summarisation.

3.5 Natural Language Processing for Summarisation

The use of NLP tools in automatic summarisation follows the same steps of that in information retrieval systems. One of the most important factors in automatic text summarisation is the process of selecting (in other word "extracting") sentences, which can be enhanced by applying NLP tools such as basic tools like stemmer and stop-words removal or more advanced tools such as co-reference resolution, discourse analysis or named-entity recognition. It has been reported that the use of NLP tools contributes to enhancing the quality of the retrieving process, the effect is more crucial in Arabic than it is for English applications [Croft et al., 2009]. Light and root stemmers in addition to standard and domain specific stop-words lists are among the tools used in automatic

text summarisation. NLP tools have been used with other models to produce robust tools for the use in information retrieval/extraction and automatic text summarisation. These tools can be used to perform many processes, which could include redundancy elimination, finding relationships and similarities between sentences, generating words to create summaries or to connect sentences in the case of abstractive summarisers.

Dimension reduction such as Latent Semantic Analysis (LSA) is a technique in natural language processing to analyse relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA has been applied to tasks that include information retrieval, information filtering and text summarisation [Deerwester et al., 1990]. In the case of Information Retrieval (IR) it is called Latent Semantic Indexing (LSI) [Yan et al., 2008]. LSA in automatic text summarisation may be able to enhance sentence selection: it finds semantic relationships between the extracted sentences rather than word by word matching relations [Hofmann, 1999].

As NLP tools are considered language dependent, the use and the effect differs from one language to another. For example the grammar of the Arabic language and the structure of the sentence is different from that of English and most of the European languages. The English language follows the Subject-Verb-Object (SVO) typology (e.g. John ate the apple), while in Arabic the typology used is Verb-Subject-Object (VSO) (e.g. Ate John the apple), in addition to that, Arabic has a rich morphology and is highly derivational [Al-Muhtaseb and Mellish, 1997].

3.6 Summary

The general architecture of any automatic text summarisation system is divided into two main modules, document selection and document summarisation. Document selection can be achieved using various tools such as a document retrieval system. The

document summarisation module differs from one summariser to another based on the summarisation approach used. The summarisation module is made of four main processes, sentence splitter, sentence matcher, sentence selector and fusion. The four processes together are responsible in generating a single or multi-document summary. Prior to the summarisation process, the data collection used needs to undergo pre-processing steps that include indexing and applying natural language processing tools, such as tokenisation, stemming and stop-words removal. This will help in finding relationships between words and sentences in order to help ranking sentences and reducing redundancy, in the case of multi-document summarisers. All the previous steps and processes are crucial for generating coherent single and multi-document summaries.

Chapter 4

Creating Resources

Researchers working on automatic summarisation require resources (sample data) in order to run their experiments. These include a collection of documents together with gold-standard summaries. These may be human generated, e.g., using crowd-sourcing or human experts, or machine generated, e.g., using machine translation tools. For some languages such as English such resources are readily available. Furthermore, numerous results have been published. This allows researchers to compare their work with that of others when judging their summariser's quality and performance. Resources, such as corpora, are important for researchers working on Arabic [Al-Sulaiti et al., 2006]. There were few such resources for Arabic summarisation when this work was started. There were neither gold-standard summaries nor published results. At the very least, to make progress on Arabic automatic summarisation, gold-standard summaries, especially human ones, were needed to be able to run automatic evaluation metrics.

Creating resources for single and multi-document text summarisation for any language requires a standard dataset and native speakers. Obstacles are then in finding documents whose use is not restricted and participants who can read and write Arabic. The problem is harder if there is little funding. For single-document summarisation,

data-sets can be obtained from websites that offer public copyrights like Wikipedia[1]. For multi-document summarisation, any data-set selected should ideally include subsets of related articles, for example news articles from multi-sources, an example for that would be the Wikinews website[2]. Perhaps the most critical part is to find native speakers of the language for which the dataset is required. Depending on the nature of the documents, the participants role can include translating the dataset and creating manual summaries. In the case of manual summaries it is appropriate to have a number of summaries for each document or set of related articles (*model summaries*) each created by a different participant. This helps avoid individual bias in the summaries. The manually translated and summarised documents need to be evaluated by other participants to provide some assurance of the translation and summarisation quality. In the Sections below, we show the creation of our publicly available Arabic single-document gold-standard summaries and our contribution to creating a multilingual multi-document parallel corpus.

This chapter describes the creation of resources for Arabic single and multi-document summarisation, including human and system gold-standard summaries. The tasks involved include translating articles from English into Arabic, summarising the translated articles and evaluating the translation and the summarisation quality. These tasks are performed by human evaluators and by using computer tools. The translation is performed by human participants and by the use of machine translation systems. The process of creating resources is crucial for us, as our work is dependent on the existence of Arabic single and multi-document summaries corpora. We believe that the resources created will help in advancing the research on Arabic natural language processing, especially in the field of single and multi-document summarisation. It could also attract researchers who are looking to enhance the summarisation evaluation tech-

[1]http://www.wikipedia.org/
[2]http://www.wikinews.org/

niques available through experimenting with the created resources. The work described here has been published in [El-Haj et al., 2010],[El-Haj et al., 2011a], [El-Haj et al., 2011c] and [Giannakopoulos et al., 2011].

4.1 Creating Resources for Single-document Summarisation

The main contribution of this work was to provide a human single-document summaries corpus with annotated summaries. The motivation behind creating Arabic resources for single-document summarisation is that good multi-document summarisation relies on solid single-document summarisation. Therefore, this work is the initial building block for creating multi-document summaries corpus later on. As observed in Chapter 2 there has been a shortage of resources for Arabic. These are required to make progress on Arabic NLP in general and summarisation in particular. Our work is mainly about multi-document text summarisation, but as a step towards that we initially investigated single-document summarisation.

In our initial work on single-document summarisation we ran a number of experiments that included automatically summarising a set of Arabic articles from Wikipedia[1], and online newspapers. The summaries generated had to be manually evaluated due to the absence of gold-standard summaries. The manual evaluation was costly and it was hard to evaluate more than a single run. Evaluation will be discussed in detail in Chapter 6.

To avoid manual evaluations and to speed up the process of measuring the quality of any of our single-document summarisers we decided to create an Arabic single-document summaries corpus. The corpus included manually created summaries that are intended to be used as gold-standard summaries when evaluating our summarisers.

[1] http://www.wikipedia.org/

To recruit a sufficient number of native users of Arabic to create human summaries we decided to use an online workforce service called *Mechanical Turk*[1] (MTurk) [Albakour et al., 2010; Alonso and Mizzaro, 2009; Kazai et al., 2011; Snow et al., 2008; Su et al., 2007]. We asked MTurk workers to provide us with a summary for a number of articles obtained from Wikipedia. The corpus which we called Essex Arabic Summaries Corpus (EASC) has been made publicly available with the aim of advancing research on Arabic single-document summarisation[2]. We are working on providing EASC through the European Language Resources Association (ELRA)[3].

4.1.1 The Document Collection

The document collection used in the creation of the single-document summaries corpus was extracted from the Arabic language version of Wikipedia and two Arabic newspapers; Alrai[4] from Jordan and Alwatan[5] from Saudi Arabia. These sources were chosen for the following reasons:

- They contain real text as would be written and used by native speakers of Arabic.
- They are written by many authors from different backgrounds.
- They cover a range of topics from different subject areas (such as politics, economics, and sports), each with a credible amount of data.

Furthermore, we could obtain the copyright holders permission to distribute the results (text from the BBC had to be excluded for this reason). The Wikipedia documents were selected by asking a group of students to search the Wikipedia website for arbitrary topics of their choice within given subject areas. The ten subject areas were:

[1] `http://www.mturk.com/`
[2] `http://privatewww.essex.ac.uk/~melhaj/corpora.htm`
[3] http://www.elra.info/
[4] `http://www.alrai.com/`
[5] `http://www.alwatan.com.sa/`

art and music; the environment; politics; sports; health; finance and insurance; science and technology; tourism; religion; and education. To obtain a more uniform distribution of articles across topics, the collection was then supplemented with newspaper articles that were retrieved from the newspaper's websites using the same queries as were used for selecting the Wikipedia articles. The total number of documents used was 153 with a total number of 18,264 words. Each document contains on average 380 words, with a minimum word-count of 116 words and a maximum of 971 words. Table 4.1 shows the corpus statistics of EASC.

Corpus Name	Essex Arabic Summaries Corpus (EASC)
Number of Documents	153
Number of Sentences	2,360
Number of Words	41,493
Number of Distinct Words	18,264
Number of Gold-standard summaries	765 (five for each document)

Table 4.1: EASC Corpus Statistics

4.1.2 Creating Manual Summaries

The corpus of extractive document summaries was generated using Mechanical Turk as follows. The documents were published as "Human Intelligence Tasks" (HITs). The assessors (workers) were asked to read and summarise a given article (one article per task) by selecting what they considered to be the most significant sentences that should make up the extractive summary. The sentences were displayed to the users as an enumerated list, the sentences were numbered so the users could select the sentence numbers they believe should be in the summary. They were required to select no more than half of the sentences in the article.

Using this method, five summaries were created for each article in the collection. Each of the summaries for a given article were generated by five different workers.

In order to verify that the workers were properly engaged with the articles, and

provide a measure of quality assurance, each worker was asked to provide up to three keywords as an indicator that they read the article and did not select random sentences. In some cases where a worker appeared to select random sentences, the summary is still considered as part of the corpus to avoid the risk of subjective bias.

Payments made to the users were dependent on the document size, ranging from \$0.06 to \$0.50 per task.

Appendix A shows the guidelines given to the MTurk workers for completing the task in addition to a HIT example.

Creating Gold-standard Summaries for EASC

As the creation of EASC corpus was based on the participants' selection of sentences, we decided to divide the participants' selection into a number of levels. Each level will form a new collection of gold-standard summaries, we call this the aggregation method [Kittur et al., 2011]. To obtain a better understanding of the impact of the aggregation method on the results of the evaluation, we constructed three different gold-standard summaries for each document based on the idea above. First of all we selected all those sentences identified by at least three of the five annotators (we call this *Level 3* summary). We also created a similar summary which includes all sentences that have been identified by at least two annotators (called *Level 2*). Finally, each document has a third summary that contains all sentences identified by any of the annotators for this document (called *All*). This last kind of summary will typically contain outlier sentences. For this reason, only the first two kinds of aggregated summaries (*Level 2* and *Level 3*) should really be viewed as providing genuine gold standards. The third one (*All*) is considered just for the purposes of providing a comparison. To illustrate that, assume we have three participants A, B and C. Assume we provided the three participants with a document made of six sentences. Following the EASC summarisa-

tion guidelines in Appendix A, the users must provide a selection of a maximum three sentences as a summary. For example, consider the following selections, where A, B and C are three random participants and the numbers are the sentences they selected: A(1,3,5), B(1,2,3), C(1,4,5). The three gold-standard summaries will be as follows: the *Level 3* summary will consist of sentence 1 only, *Level 2* summary will contain sentences 1, 3 and 5. And finally the *All* summary will contain sentences 1,2,3,4 and 5.

Concluding Remarks

In addition to addressing the shortage of relevant resources for Arabic natural language processing, this work also demonstrated the application of Mechanical Turk to the problem of creating natural language resources. The primary output of this project was a corpus of 765 human-generated summaries, which is now available to the community[1]. The corpus will be made available through the European Language Resources Association (ELRA)[2]. The number of generated human summaries was dependent on the available fund. We believe that the process of creating the EASC corpus could inspire the creation of more human generated single-document summaries. To simplify the use of EASC corpus when evaluating summarisers, the file names and extensions are formatted to be compatible with current evaluation systems such as ROUGE and AutoSummENG. It is also available in two character encodings, UTF-8 and ISO-8859-6 (Arabic). We use this corpus in our experiments on single-document summarisation (Chapter 5).

[1] `http://privatewww.essex.ac.uk/~melhaj/corpora.htm`
[2] http://www.elra.info/

4.2 Creating Resources for Multi-document Summarisation

Our experience with using Mechanical Turk to create single-document summaries suggests that crowd sourcing is an economical and efficient way to create single document summaries. However, using the same approach to create a multi-document summaries corpus is difficult as it will be costly in the matter of time and money. It would require asking users to read numerous related articles and then summarise them. For these reasons, it seems that using Mechanical Turk for creating multi-document gold-standard summaries is not viable or feasible in the context of the current work.

For the above reasons, we did not use MTurk for creating the multi-document corpus, and instead applied different approaches. In the first approach we used machine translation to translate into Arabic a freely available English dataset that has already been used and tested for multi-document summarisation tasks. In the second approach we manually created an Arabic multi-document summaries corpus. This funded work was undertaken as part of our role in organising the MultiLing Summarisation Pilot[1], part of the Text Analysis Conference[2] (TAC).

4.2.1 Automatic Creation of a Multi-document Summaries Corpus

In this approach we used machine translation to automatically create an Arabic multi-document summaries corpus. The machine translation was used to translate an English dataset into Arabic. The output of the translation process is an Arabic dataset of related news articles. The dataset used in the machine translation process was the DUC–

[1] http://users.iit.demokritos.gr/~ggianna/TAC2011/MultiLing2011.html
[2] http://www.nist.gov/tac/

2002 dataset provided freely by the National Institute of Standards and Technology[1] (NIST) through the Document Understanding Conference[2] (DUC).

DUC–2002 is an English dataset that contains 567 articles in addition to 1111 gold-standard (model) summaries. NIST produced 60 reference sets. The reference sets were produced using data from the TREC question-answering track in TREC-9[3]. Each set had between five and fifteen documents, with an average of ten documents. The documents are of at least ten sentences long with no maximum length. Table 4.2 shows the corpus statistics of the DUC–2002 Arabic translation dataset.

Corpus Name	DUC–2002 (Arabic Translation)
Number of Documents	567
Number of Sentences	17,340
Number of Words	199,423
Number of Distinct Words	19,307
Number of Reference Sets	59
Documents Per Reference Set	10 on average
Number of Gold-standard summaries	118 (two for each reference set)

Table 4.2: DUC–2002 Arabic Corpus Statistics

The topics of each of the reference sets consisted of one of the following categories. Each category had an equal number of reference sets[4].

- Documents discussing a single natural disaster created within a seven day window.
- Documents about a single event in any domain and created within at most a seven day window.
- Documents about multiple distinct events of a single type with no limit on the time window.

[1] `http://www.nist.gov/index.html`
[2] `http://duc.nist.gov/`
[3] `http://trec.nist.gov/pubs/trec9/t9_proceedings.html`
[4] `http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html`

- Documents presenting biographical information mainly about a single individual.

The model summaries included both single and multi-document abstract/extract summaries. We are only interested in the multi-document extractive summaries. There are two multi-document extractive summaries for each reference set of 400 and 200 words respectively. The 200 word summaries were extracted from the 400 words ones. No partial sentences were used.

Arabic Translation of DUC–2002 Dataset

The DUC–2002 dataset was automatically translated into Arabic on a sentence-by-sentence basis. To automatically translate the dataset we used the Java version of Google translate API[1]. A total of 17,340 sentences were translated. Google translate has its limitations on the size of text and the number of requests made in a period of time. In order to accommodate this issue we set a Java 500 milliseconds sleep-time. This made it possible for the translation task to continue without interruption messages or time delay penalty from Google.

A key reason for selecting the DUC–2002 dataset is that this data has been used in multi-document summarisation tasks and there are many published results that used this dataset. This provides us with data with which we can compare the results of our summariser when using an Arabic corpus derived from the same data. The produced Arabic dataset was used in the multi-document summarisation experiments presented in Chapter 5.

[1] `http://code.google.com/p/google-api-translate-java/`

4.2.2 Manual Creation of a Multi-document Summaries Corpus

In this section we discuss our final approach in creating Arabic resources for multi-document summarisation. In contrast to the previous section, the work here used human participants to manually create the resources. The manual process included translating, validating and summarising documents written in English.

The process of manually creating an Arabic multi-document corpus was part of our organisation for the TAC–2011 MultiLing Summarisation Pilot[1].

The participants are responsible for translating and summarising an English test collection into six languages including: Arabic, Hindi, French, Czech, Greek and Hebrew. The test collection was based on WikiNews texts[2]. The source documents contained no meta-data or tags and were represented as UTF–8 plain text files. The test collection contained 100 articles divided into ten reference sets, each contained ten related articles discussing the same topic. The original language of the dataset was English. Table 4.3 shows the corpus statistics of the TAC–2011 Arabic dataset.

Corpus Name	TAC–2011 MultiLing (Arabic)
Number of Documents	100
Number of Sentences	1,573
Number of Words	30,908
Number of Distinct Words	9,632
Number of Reference Sets	10
Documents Per Reference Set	10 on average
Number of Gold-standard summaries	30 (three for each reference set)

Table 4.3: TAC–2011 Arabic Corpus Statistics

The preparation of the Arabic corpus for the TAC–2011 MultiLing Summarisation Pilot is our focus of interest here. A total of twelve people participated in translating the English corpus into Arabic, summarising the set of related Arabic articles, validating

[1] http://www.nist.gov/tac/2011/Summarization/index.html
[2] http://www.wikinews.org/

the translation and evaluating the summarisation quality. The participants were paid using Amazon vouchers. The amount of the vouchers varied depending on the task performed. The total amount of Amazon vouchers paid to the participants was £250 where three of the participants volunteered to do the tasks. The participants have been selected based on their proficiency in Arabic and each one of them must be an Arabic native speaker. The participants are studying, or have finished a university degree in an Arabic speaking country. The participants age range between 22 and 64 years old.

The participants translated the English dataset into Arabic. For each translated article another translator should validate the translation and fix any errors. For each of the translated articles, a number of three manual summaries are created by three different participants (human peers). Amid the summarisation process the participants should evaluate the quality of the generated summary by assigning a score between one (unreadable summary) and five (fluent and readable summary). No self evaluation is allowed. Appendix B shows the detailed guidelines required by the participants. The created corpus was used in our experiments with multi-document summarisation (Chapter 5).

The average time for reading the English news articles by the Arabic native speaker participants was four minutes. The average time it took them to translation those articles into Arabic is 25 minutes, and to validate each of the translated Arabic articles the participants took six minutes on average. For the summarisation task the average time for reading the set of related articles (ten articles per each set) was seventeen minutes. The average time for the summarisation process of each set was 24 minutes. The participants were also asked to evaluate the summaries they generated. However, no participant should evaluate his own work. The participants were aware that the summaries they are about to evaluate are human summaries generated by native Arabic speakers.

We consider that the creation of the TAC-2011 MultiLing corpora is an important

contribution to automatic summarisation. For the past eleven years participation to DUC and later to TAC workshops was limited to English summarisers only due to the absence of any other summarisation corpora. For Arabic researchers the created corpus could help in advancing the work in many fields that includes: automatic summarisation, question answering and automatic translation. The corpus is freely available by filling the forms required by NIST[1].

4.3 Summary

Resource creation plays an important role in the advance of Arabic single and multi-document summarisation. The created Arabic gold-standard summaries are needed to evaluate the generated summaries by our Arabic single and multi-document summarisers. Therefore, three automatic summaries corpora have been created. The first created corpus was the Essex Arabic Summaries Corpus (EASC), a single-document Arabic extractive summaries. The corpus was created manually using participants from Mechanical Turk – an online workforce. The approach demonstrated the application of Mechanical Turk to the problem of creating natural language resources. The created corpus contains 765 human-generated summaries. The second corpus is a multi-document summaries created using a machine translation tool to translate, into Arabic, the DUC–2002 dataset. The approach used introduced a cost-effective solution to the problem of limited Arabic resources. The third corpus, MultiLing dataset, is a multi-document summaries corpus. The corpus was created by native Arabic speakers who participated by manually translating English documents into Arabic. The participants manually summarised the translated dataset and evaluated the created summaries. The MultiLing corpus could benefit researchers on Arabic natural language processing, especially those working on multi-document summarisation.

[1] `http://www.nist.gov/tac/2011/Summarization/index.html`

Chapter 5

Summarisation Methodology

This chapter presents methods and implementations of extractive single-document and multi-document summarisation, the summarisers description and the experimental setup. We consider single-document summarisation first, as it forms a backbone for our work on multi-document summarisation. Single-document extractive summaries were produced using a number of methods, including query-based, concept-based and generic summarisers. Section 5.1 shows the work on single-document summarisation. The section is divided into two main subsections. Subsection 5.1.1 considers the use of query-based and concept-based single-document summarisation. Subsection 5.1.2 shows the use of generic summarisation by applying language–dependent and language–independent techniques.

Section 5.2 shows the work on multi-document summarisation. The section is divided into three main subsections. Subsection 5.2.1 considers the use of statistical and semantic models to summarise and eliminate redundancies. Subsection 5.2.2 shows the use of clustering to enhance the summaries quality by providing a novel cluster-based redundancy elimination technique. The subsection also discusses our participation in the TAC–2011 competition using language–independent multi-document summarisers. The DUC–2002 dataset was used in all the experiments of the first two subsections.

Subsection 5.2.3 discusses our language–dependent summarisers using Natural Language Processing (NLP) tools. We used TAC–2011 Arabic and English datasets in all the experiments of this section.

5.1 Single-document Summarisation

5.1.1 Query and Concept based Summarisation

In earlier work, as part of my master degree thesis, two single-document extractive Arabic summarisers were developed: the Arabic Query-Based Text Summarisation System (AQBTSS) and the Arabic Concept-Based Text Summarisation System (ACBTSS) [El-Haj, 2008]. Here we describe them, and present an evaluation of their performance. The summarisers' primary data source was a collection of Arabic articles extracted from Wikipedia, a free online encyclopedia[1].

5.1.1.1 Arabic Summarisers: AQBTSS and ACBTSS

AQBTSS attempts to provide a reasonable summary for a document that is relevant to a specific user's query. ACBTSS attempts to provide a summary of a document that is relevant to a chosen concept, as represented by a set of words instead of a user's query.

The concepts used in the ACBTSS concept-based summariser include: art and music, the environment, politics, sports, health, finance and insurance, science and technology, tourism, religion and education. Some concepts for Arabic document classification are given by [Khreisat, 2006]. In addition to these, we included concepts that are commonly used by many Arabic newspapers. The words used to represent a concept in ACBTSS were selected by way of a statistical analysis of 10,250 Arabic articles from different Arabic newspapers, with approximately 850 documents for each

[1] `http://www.wikipedia.org/`

of the above mentioned concepts. Essentially we selected the most frequent terms for each chosen category, and removed stop-words (the stop-words list consisted of the 300 most frequent words in the 10,250 articles).

Figure 5.1 depicts the general flow diagram of AQBTSS and ACBTSS. Both summarisers consist of two major modules. The first is the *Document Selection* module. In this module the users search the document collection to find documents that satisfy their query. The initial selection is performed using a simple concordance system. The users then select a document from this initial selection for summarisation.

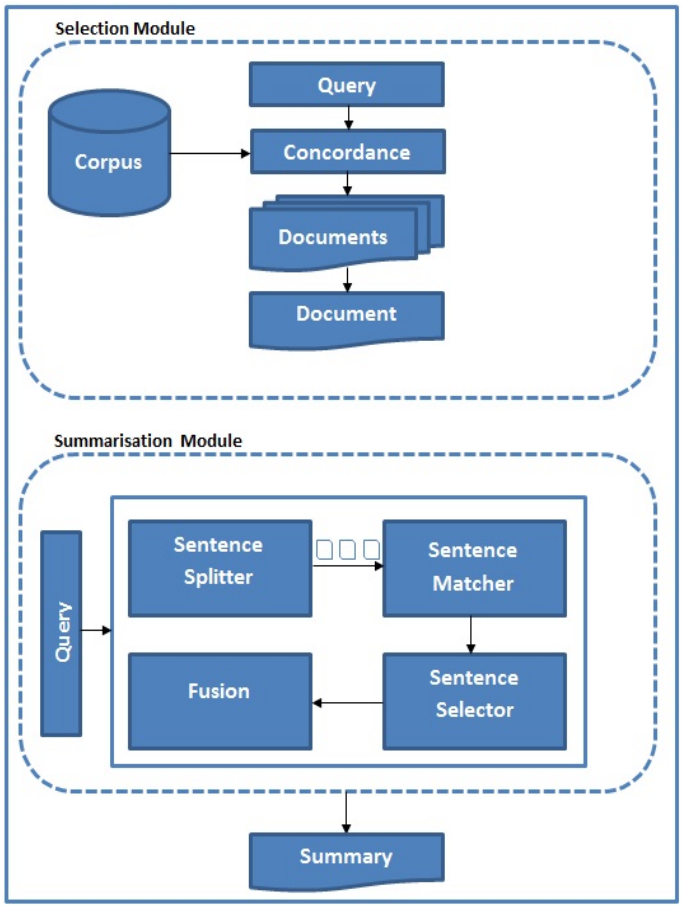

Figure 5.1: AQBTSS and ACBTSS Diagram

In the *Document Summarisation* module, the summariser starts by splitting the documents into sentences. AQBTSS and ACBTSS differ in how they perform sentence

matching. In AQBTSS each sentence is compared against the user query to find relevant sentences. This is the same query that was used in the document selection module. The ACBTSS sentence matcher ignores the user query that was used to select the documents. Instead, each sentence is matched against a set of keywords that represent a given concept.

To rank the matched (extracted) sentences we adopted the Vector Space Model (VSM) [Salton et al., 1975]. The weighting scheme based on the VSM makes use of two measures: Term Frequency (TF) and Inverse Document Frequency (IDF), see Section 3.1.

5.1.1.2 Experimental Design

We tested our summarisers using a set of forty queries to retrieve a set of documents. The summarisers generated a summary for each returned document. A group of 1,500 users participated in evaluating the readability of the generated summaries. The users are located in Jordan and they are all native Arabic speakers.

The Document Collection

The evaluation of the two single-document summarisers was performed using a document collection that consisted of forty documents selected from 251 articles on various subjects. The initial collection of 251 articles was obtained by asking a group of students to search the Wikipedia website for articles using their own queries. The result of the process was a collection of articles and their associated queries. The concordance system was then used to select the forty documents (and their associated queries), to be used in the experiments, from the initial 251 articles. These forty articles were the first articles retrieved from the initial 251 using a concordance system using topics from TREC[1]. The total size of the collection was 95,933 words. The average size of each

[1]http://trec.nist.gov/

article was 378 words. The work on AQBTSS and ACBTSS and their evaluation was published in El-Haj et al. [2009].

5.1.2 Generic Summarisation

The previous experiments on single-document summarisation used a query or conceptual keywords in the summarisation process. In this section we discuss generic extractive single-document summarisers for Arabic, called "*Gen–Summ*" and "*LSA–Summ*". The reason behind assessing "*Gen–Summ*" and "*LSA–Summ*" is to provide a user independent summariser that is capable of extracting relevant sentences without the need for any user intervention. *Gen–Summ* is similar to query-based summarisers except that the query is replaced by the document's first sentence. *Gen–Summ* uses VSM [Salton et al., 1975] to select the sentences that are most similar to the initial selected sentences. *LSA–Summ* is similar to *Gen–Summ* except that the vector space is transformed and reduced by applying Latent Semantic Analysis (LSA) [Deerwester et al., 1990] to analyse relationships between document's sentences and the terms they contain by producing a set of concepts related to the documents and terms. Both summarisers provide a summary with no more than 50% of the document's words count.

5.1.2.1 Experimental Design

Rather than using a query or concept the generic summarisers use the first sentence to identify the topic to be summarised. The justification for selecting the first sentence as being representative of the relevant topic is based on the belief that in many cases, especially news articles, the first sentence tends to contain information about the content of the entire article [Radev et al., 2004], [Fattah and Ren, 2008] and [Yeh et al., 2008]. The fact it is often included in extractive summaries generated by more sophis-

ticated approaches provides some corroboration [Baxendale, 1958] and [Katragadda et al., 2009].

For the evaluation of *Gen–Summ* and *LSA–Summ* we used EASC – the Essex Arabic Summaries Corpus (Chapter 4). The participants who created the EASC corpus generated the summaries by reading the documents without being given queries to guide the summarisation. This is a better match for evaluating summarisers that do not use queries or concepts.

The *Gen–Summ* and *LSA–Summ* results were compared to the AQBTSS summariser and "Sakhr" – a commercial online Arabic text summariser available on the web[1]. It should be noted that Sakhr was only a beta release at the time we performed our experiments. Sakhr consists of a set of text-mining tools to identify the most relevant sentences within a document and displays them in the form of a prioritised list of key sentences.

As a baseline for the evaluation, we created a simple summariser "Baseline–1" that takes the first sentence of a document as a summary. The results of this summariser were compared to the *Gen–Summ*, *LSA–Summ* and Sakhr single-document summarisers.

The *Gen–Summ* and *LSA–Summ* experiment was divided into two phases, once with the use of an Arabic light stemmer (see Section 3.2.1) and stop-words removal (see SubSection 5.1.1) and the other with the absence of a stemmer. The Arabic light stemmer was not used in AQBTSS and Sakhr. It was not possible to apply stemming on Sakhr as it is not an open source summariser, and therefore we did not apply stemming on AQBTSS, to be consistent when comparing both summarisers. The evaluation of the summarisers presented in this section will be shown in Chapter 6. Section 6.1 presents the human evaluation while Section 6.2 presents the automatic evaluation.

[1] `http://www.sakhr.com/`

5.2 Multi-document Summarisation

5.2.1 Statistical and Semantic Summarisation

In this section we introduce different extractive multi-document summarisers. The summarisers read a set of related articles and extract sentences that are similar to the first sentence of each article. Statistical and semantic methods have been applied which include dice's coefficient, Vector Space Model and Latent Semantic Analysis. For the experiments of this section we used the English and the parallel Arabic translation of the DUC–2002 extractive multi-document dataset (see Chapter 4).

5.2.1.1 Experimental Setup

The Document Collection

The document collection used in this experiment was the DUC–2002 dataset and its translation into Arabic. As the focus of this work is on Arabic multi-document summarisation, we had to create this translated corpus. The translation into Arabic of the DUC–2002 dataset involved the documents only. The gold-standard summaries were not translated. As the translation process produced a parallel Arabic/English dataset and since the gold-standard summaries are extractive ones, we were able to retrieve the Arabic translation of the gold-standard summaries sentences.

The Summarisers

We have used three extractive multi-document summarisers for both Arabic and English languages. The summarisers are called *VSM–Sum*, *LSA–Sum* and *Dice–Sum*. The summarisers provide a single summary for each of the reference sets of the DUC–2002 dataset. In each of the three summarisers, the sentence selection/extraction process is done to each article separately from the others. Then based on the cosine measure

using the Vector Space Model (VSM) the summarisers select the top two sentences from each article before the redundancy elimination phase starts. VSM was used to select the sentence that has the maximum sum of weight when compared to all the other sentences in an article. For each article the selected sentences will always include the first sentence and the most relevant sentence to it, using the VSM ranking. We did not select more than two sentences from each article to avoid exceeding the limit of the generated summaries. The justification for selecting the first sentence as being representative of the relevant topic is based on the belief that in many cases, especially news articles, the first sentence tends to contain information about the content of the entire article [Radev et al., 2004], [Fattah and Ren, 2008] and [Yeh et al., 2008]. The fact it is often included in extractive summaries generated by more sophisticated approaches provides some corroboration [Baxendale, 1958] and [Katragadda et al., 2009].

Dealing with multi-documents we usually end up having redundant sentences in the generated summaries. To address this problem we have applied a number of statistical and semantic techniques to eliminate redundancies from the generated summaries. The summarisers differ in the redundancy elimination technique used. The description of each of the summarisers is given as follows.

VSM–Sum summariser used VSM to find the similarity between a pair of sentences. If the similarity measure reaches a certain threshold then the two sentences are considered redundant and only one of them is kept (the sentence that has the maximum sum of weight when compared to all the other sentences in an article). The process of removing redundancy using VSM is done by selecting the sentences extracted by VSM–Sum one by one and comparing them against each other. The similarity measure applies on a threshold. In our case we took the difference between the weights of the two vectors. If the difference is less than 0.2 we consider this sentence redundant. The threshold was established experimentally. One of the flaws of using this method to eliminate redundancy is that the

elimination process could be aggressive and we might end up with a one sentence summary [Salton et al., 1975].

LSA–Sum used Latent Semantic Analysis (LSA) [Deerwester et al., 1990; Dumais et al., 1988] to find relationships between the document's sentences and the terms they contain. This was done by producing a set of concepts related to the documents and terms. The concepts were used to find the similarity between two sentences and thus eliminate redundancy. A threshold of similarity was assigned where a sentence is considered redundant if this threshold of similarity was exceeded. Using trial and error again, the threshold of similarity used was 0.2.

Dice–Sum used Dice's coefficient [Manning and Schütze, 1999] to find similarities between the extracted sentences using "bag of words". Dice's coefficient ranges between zero and one. The difference between Dice's coefficient and VSM is that Dice's coefficient is not a proper distance metric (see Section 6.2). The threshold of similarity used was 0.5 using trial and error.

The Summarisation Process

We used the Arabic and English versions of VSM–Sum, LSA–Sum and Dice–Sum in the summarisation process. The process was done in two phases. In the first phase, we summarised the 59 English reference sets (567 articles) of the DUC–2002 dataset using the English version of our multi-document summarisers. ROUGE and the English gold–standards were used to evaluate those summaries. In the second phase, we summarised the Arabic parallel translation version of DUC–2002 dataset using our Arabic multi-document summarisers. As we have only English gold-standard summaries we could not evaluate the Arabic summaries directly. To address this problem we selected the parallel English sentences of each Arabic summary. Having this we were then able

to evaluate these equivalent English summaries using ROUGE and the English gold–standards. The ROUGE evaluation was performed having a confidence interval of 95% which was believed to be equivalent to human evaluation [Lin, 2004].

Section 7.1 illustrates the evaluation results of the statistical and semantic summarisation experiments.

5.2.2 Cluster-based Summarisation

In this section we introduce Arabic and English cluster-based multi-document summarisation techniques. The DUC–2002 dataset was used, both in English and Arabic. For all our experiments we used a generic multi-document summariser that was implemented for both Arabic and English (using identical processing pipelines for both languages). The only difference here is that we did not use the redundancy elimination models presented earlier (see Section 5.2.1). Instead we use clustering for redundancy elimination. At the end of this section we show the experimental setup of our participation at the TAC–2011 MultiLing workshop.

The following subsections describe the clustering methods employed in our experiments, the actual summarisation process and the experimental setup. We explored clustering for two applications, first to aid sentence selection, second to aid redundancy elimination in an initial selection of sentences.

5.2.2.1 Experiment 1: Clustering all Sentences

In this experiment we treat all documents to be summarised as a single bag of sentences. The sentences of all the documents are clustered using different numbers of clusters. Then we select a summary from the clusters using two alternative approaches:

1. Select sentences from the biggest cluster only (if there are two we select the first biggest cluster based on the clusters order).

2. Select sentences from all clusters.

The intuition for the first approach is the assumption that a single cluster will give a coherent summary all centred around a single theme, whereas the second approach is expected to result in summaries that contain more aspects of the topics discussed in the documents and therefore yield a summary that gives a broader picture.

5.2.2.2 Experiment 2: Clustering for Redundancy Elimination

Redundancy elimination is an important part of automatic summarisation. A summary that contains very similar sentences drawn from different documents is not ideal. In Section 5.2.1 we used redundancy elimination algorithms applied to an initial set of selected sentences. We believe that the use of VSM, LSA and Dice's coefficient may have led to some important sentences being left out. The expectation was that clustering of the pre-selected sentences has the potential to improve the quality of the summarisers. Instead of deciding on which sentence to keep and which sentence to remove, we cluster the sentences into a different number of clusters. Then we select sentences from the clusters using the sentence selection approaches mentioned in Subsection 5.2.2.1. Using this method we select the sentences that are closer to the centroid (important sentences) to produce high quality summaries.

5.2.2.3 Experimental Setup

The algorithm used in the experiments is K-means clustering [Lloyd, 1982]. This is a partitional centroid-based clustering algorithm. The algorithm randomly selects sentences as the initial centroid for each cluster. The K-means algorithm then iteratively assigns all sentences to the closest cluster, and recalculates the centroid of each cluster, until the centroids no longer change. For our experiments, the similarity between a sentence and a cluster centroid is calculated using the standard cosine measure applied to tokens within the sentence.

We run K-means clustering using different number of clusters 1, 2, 5, 10, 15 and 20. Clustering using a single cluster essentially results in a list of sentences which can be ranked according to similarity to the centroid of all sentences.

Summary sentences are being selected from the clusters using two selection methods:

1. Select the *first* sentence of each cluster
2. Select all the sentences in the *biggest* cluster.

The ranking of sentences is done according to similarity to the centroid. The biggest cluster is defined as the one comprising the largest number of sentences. In the resulting summary we keep the order of sentences as they appear in the clusters (i.e. a sentence very similar to the centroid appears earlier on in the summary than the one that is less similar).

Once redundancy elimination is done, we summarised the Arabic and English versions of the DUC–2002 following the same summarisation process described in Section 5.2.1.

Note that in our experiments we do not trim the resulting summary to a particular length. ROUGE will do this automatically for summaries that are too long by reading summaries up to a limit that we specify. Alternatively, we could produce a summary that does not exceed a fixed maximum length.

Where appropriate, we determined significant differences by performing pairwise *t.tests* ($p<0.05$).

5.2.2.4 TAC–2011 MultiLing Summarisation Experiment

This subsection shows the experimental setup and the summarisation process of our participation at the TAC–2011 MultiLing summarisation competition.

The Text Analysis Conference (TAC) is one of the leading conferences on automatic summarisation, groups participating in this conference have the chance of comparing

their summarisation systems to others using different or similar techniques. Participating in TAC–2011 was among the contributions of this thesis to measure the performance of our summarisers by being compared to other state-of-the-art summarisers. We participated in the MultiLing Summarisation workshop, which is a multilingual task for multi-document summarisation systems. We were the organisers of the Arabic language and we participated with two multi-document language–independent summarisers, one for Arabic and the other for English. The dataset used in the MultiLing Summarisation workshop was the TAC–2011 Multilingual multi-document dataset, which is available in 7 languages including Arabic and English. See Chapter 4 for the details on the preparation and the creation of the dataset. In our participation at the TAC–2011 MultiLing workshop, we only considered the clustering approach that treats all documents to be summarised as a single bag of sentences. We clustered the sentences using a single cluster (see Subsection 5.2.2.1). We found that this selection method performed well on average (as we will see later in Section 7.2).

Note, that in this experiment we trimmed the resulting summary to a particular length. As indicated in the MultiLing workshop, the acceptable limits for the word count of a summary were between 240 and 250 words (inclusive) [Giannakopoulos et al., 2011].

Section 7.2 shows the evaluation results of the cluster-based summarisation experiments in addition to the results of our participation at TAC–2011 MultiLing workshop.

5.2.3 NLP Tools Summarisation

In this work we applied a set of natural language processing (NLP) tools to examine whether they can be used to improve summarisation quality. The dataset used in this experiment was the TAC–2011 MultiLing dataset (see Subsection 5.2.2.4). The experiment was applied to the Arabic and the English datasets as we wanted to see

the effect of using NLP tools and compare this to the work done in Subsection 5.2.2.4. The set of NLP tools used in this experiment were:

Arabic light stemmer, the proposed stemmer removes the most common prefixes and suffixes [El-Haj et al., 2009].

Arabic root extractor stemmer, this stemmer was implemented by Khoja and Garside [1999], it is a root-based stemmer that removes suffixes, infixes and prefixes and uses pattern matching to extract the roots.

Arabic standard stop-word list, the list contains the most common stop-words in the Arabic language (i.e. words that occur frequently) [Khoja and Garside, 1999; Larkey et al., 2002].

Arabic domain specific stop-word list, the list was created using the TAC-2011 Arabic corpus. To create this list we followed the work by [Fox, 1989], which shows the creation of stop-word list for general text.

English Porter stemmer, which is a commonly used stemmer for removing morphological and inflectional endings from words in English [Porter, 1997].

English LOVINS stemmer, is a single pass, context–sensitive, longest match stemmer that was developed by [Lovins, 1968].

English standard stop-word list, a list that contains the most common English stop-words [Fox, 1989].

English domain specific stop-word list, is the English stop list of the TAC–2011 English corpus based on the work by [Fox, 1989].

5.2.3.1 Experimental Setup

The dataset used in our experiments with NLP tools was the TAC–2011 MultiLing dataset. As mentioned before, the dataset comes in seven languages. In this experiment we are only interested in the Arabic and English versions of the dataset. To compare the results of this experiment with the results of TAC–2011 MultiLing workshop we will apply the same summarisation techniques used in our participation at TAC–2011 summarisation workshop (Section 5.2.2.4). The motivation behind this experiment is the fact that TAC–2011 MultiLing workshop guidelines required the summarisers to be language–independent, where it is not possible to apply NLP tools such as stemmers or stop-words removals. In this experiment we are trying to test the effect of using NLP tools on the quality of the generated summaries.

5.2.3.2 Arabic Summarisation with NLP Tools

The following combinations of the tools were used, each combination is a separate experiment. Later we are going to discuss the results and their comparison with the previous ones:

1. *Standard Stop List Only*, we applied only the stop-words removal process using the Arabic standard stop-words list.

2. *Domain Specific Stop List Only*, we used the list we created to remove stop-words prior to the sentence-extraction process.

3. *Light–Stemmer Only*, the light stemmer was applied to the sentences prior to the sentence-extraction process.

4. *Root–Stemmer Only*, we used a root stemmer to extract the very basic form of the words in the sentences prior to the sentence-extraction process.

5. *Standard Stop List and Light–Stemmer*, the experiment here was to test the combined effect of using both NLP tools (1. + 3.).

6. *Domain Specific Stop List and Light–stemmer*, here we applied (2. + 3.) together.

7. *Standard Stop List and Root–stemmer*, here we applied (1. + 4.) together.

8. *Domain Specific Stop List and Root–stemmer*, (2. + 4.) were applied together.

9. *Standard and Domain Specific Stop Lists*, we applied a combination of the two stop-word lists we have for Arabic (1. + 2.).

5.2.3.3 English Summarisation with NLP Tools

For the English language we followed the same steps as for the Arabic experiments. The main difference was the use of English specific NLP tools. The experiments we ran for the English language are as follows:

1. *Standard Stop List Only*, we applied the stop-words removal process using the English standard stop-words list.

2. *Domain Specific Stop List Only*, we used the list we created to remove stop-words prior to the sentence-extraction process.

3. *Porter–stemmer Only*, the Porter stemmer was applied to the sentences prior to the sentence-extraction process.

4. *Lovins–Stemmer Only*, Lovins' stemmer was applied to the sentences prior to the sentence-extraction process.

5. *Standard Stop List and Porter–Stemmer*, the experiment here was to test the combined effect of using both NLP tools (1. + 3.).

6. *Domain Specific Stop List and Porter–Stemmer*, here we applied (2.+3.) together.

7. *Standard Stop List and Lovins–Stemmer*, here we applied (1. + 4.) together.

8. *Domain Specific Stop List and Lovins–Stemmer*, here we applied (2.+4.) together.

9. *Standard and Domain Specific Stop Lists*, we applied a combination of the two stop-word lists we have for English (1. + 2.).

Section 7.4 shows the evaluation results of the summarisation with NLP tools experiments.

5.3 Summary

The work on single-document and multi-document summarisation was systematic where we applied different approaches, methods and techniques for summarising single and multiple documents, in addition to eliminating redundancy from multiple documents. We applied those approaches both on Arabic and English following similar pipeline. The work on Arabic multi-document summarisation has not been done before and provides novel techniques for summarising Arabic documents and eliminating redundancy. The experiments with single-document summarisation provide the basic for developing the underlying methods in multi-document summarisation. We used different approaches for Arabic single-document summarisation including query-based, concept-based and generic summarisation. For the work on multi-document summarisation, we implemented a cluster-based summarisation and participated in one of the leading conferences on multi-document summarisation. We showed a novel cluster-based technique for redundancy elimination. We applied language–dependent approaches that included the use of basic natural language processing tools (stemmer and stop-word list) and semantic models such as the latent semantic analysis model. The work shown investigated many tools, techniques and methods for both Arabic single-document and multi-document summarisation that might help advancing the work on these fields and

on Arabic NLP in general. The selection of which experiment to apply or what algorithm to use was based on historical experience of discovering what resources were not available and had to be found. We tackled many areas in automatic summarisation including the creation of resources, redundancy elimination, finding similarities and extracting sentences. This could help researchers working on Arabic summarisation and NLP to improve the work by extending the experiments and applying more techniques, methods and algorithms and comparing their results to what we have achieved already.

Chapter 6

Single-document Summarisation Evaluation

This chapter shows the evaluation results of the work done on single-document summarisation presented in Chapter 5. Both human generated and automatically generated corpora were used in the evaluation. The evaluation results of the single-document summarisation experiments are presented and compared with the results of other summarisation techniques and systems in the literature. We also show details of the evaluation metrics used. The evaluation process is divided into two sections, human and automatic evaluation. We report the evaluation results of the human evaluation in Section 6.1, where a total of 1,500 users participated in manually evaluating summaries. We also show the results of the automatic evaluation metrics in Section 6.2. These include ROUGE, AutoSummENG and Dice's Coefficient.

6.1 Human Evaluation

In this section we show the human evaluation of the query-based and concept-based extractive single-document summarisers presented in Section 5.1. The human evalua-

tion was applied to AQBTSS, ACBTSS and Sakhr only as at the time of the human evaluation process the *Gen–Summ* and *LSA–Summ* summarisers were not yet developed. A total of 1,500 Arabic native speakers from different backgrounds participated in the evaluation. Each participant was given a document with two summaries, one generated by AQBTSS and the other by ACBTSS. The summaries were given to the participants in random order. Each participant was asked to read the document and its summaries and then to evaluate each summary on a five-point Likert scale [Dang, 2007]. The possible judgements, their scores, and our interpretations are given in Table 6.1.

Evaluation	**Score**	**Interpretation**
V. Poor	0	The summary is very poor and is not related to the document at all.
Poor	1	The summary is poor as the core meaning of the document is missing.
Fair	2	The user is somehow satisfied with the result, but expected more.
Good	3	The summary is readable and it carries the main idea of the document.
V. Good	4	The summary is very readable and focuses more on the core meaning of the document. The user is happy with the results.

Table 6.1: Evaluation Scale

Five groups of 300 users participated in evaluating the summarisers. The users vary in their ages and educational levels as shown in Table 6.2. There were roughly the same profiles of users in each group.

We assume that there may be some correlation between the ages of the participants and their linguistic skills. The variation in backgrounds and subjects would arguably give rise to different expectations about the likely quality of automatic summarisation of Arabic. We anticipated that some groups would be more familiar with the computational issues than others.

All the 1,500 participants evaluated the AQBTSS and ACBTSS summarisers. The

Group	Description
Humanities students	Third and fourth year Humanities students at the University of Jordan.
Arabic Literature students	Third and fourth year Arabic literature students at the University of Jordan.
Computer Science students	Various levels Computer Science students at the University of Jordan.
K-12 school students	9th and 10th grade school students studying in Amman, Jordan.
K-12 school teachers	K-12 school teachers with different specialities teaching in Amman, Jordan.

Table 6.2: Participants User Groups

Sakhr summariser was evaluated by the group of computer science students only. This is the group which gave the lowest scores when evaluating AQBTSS and ACBTSS (i.e., the most critical user group). The participants were not told the source of the summaries generated by Sakhr in order to avoid subjective bias.

6.1.1 Human Evaluation Results

Table 6.3 and Table 6.4 show the human evaluation results for AQBTSS and ACBTSS respectively. Table 6.5 presents the mean scores for the two systems, together with the p value from the student t.test. A standard pairwise t.test ($p<0.05$) was performed to determine significance, by testing each group with 300 observations on both systems.

Group	Scores				
	0	1	2	3	4
K-12 School Teachers	0.00%	2.00%	7.67%	47.33%	43.00%
Arabic Literature Students	0.00%	4.00%	11.67%	46.33%	38.00%
Humanities Students	0.33%	5.00%	14.00%	57.67%	23.00%
K-12 School Students	0.67%	3.33%	19.33%	39.33%	37.33%
Computer Science Students	1.67%	7.00%	24.00%	44.00%	23.33%
Average	0.53%	4.20%	15.40%	46.93%	32.93%

Table 6.3: Overall gradings of the AQBTSS Summariser

The significance test showed that all user groups apart from the Humanities students

Group	Scores				
	0	1	2	3	4
K-12 School Teachers	0.67%	5.00%	21.33%	38.67%	34.33%
Arabic Literature Students	1.00%	7.33%	29.67%	33.33%	28.67%
Humanities Students	1.00%	4.67%	18.00%	49.00%	27.33%
K-12 School Students	0.67%	6.33%	24.33%	42.00%	26.67%
Computer Science Students	2.33%	16.00%	35.67%	30.33%	15.67%
Average	1.13%	7.87%	25.80%	38.67%	26.53%

Table 6.4: Overall gradings of the ACBTSS Summariser

gave significantly higher ratings for the query-based summariser AQBTSS than the query-independent concept-based summariser ACBTSS (Table 6.5).

Group	Mean (ACBTSS)	Mean (AQBTSS)	p
Humanities Students	2.97	2.98	0.44
K-12 School Students	2.877	3.093	0.001
K-12 School Teachers	3.01	3.313	2.69E-06
Computer Science Students	2.41	2.803	4.59E-07
Arabic Literature Students	2.813	3.183	1.95E-07

Table 6.5: AQBTSS and ACBTSS t.test Results

Figure 6.1 shows the results of evaluation obtained from Sakhr compared to those we observed for AQBTSS (see Table 6.3) for the computer science students. This group was the one that assigned the lowest average score to both AQBTSS and ACBTSS.

In the case of AQBTSS (Table 6.3), the K-12 teachers gave on average the highest gradings followed by the group of students majoring in Arabic literature. The K-12 teachers' gradings on average were significantly higher than those of any other group. The lowest gradings were awarded by the Computer Science students. In the case of ACBTSS (Table 6.4), the Humanities students gave on average the highest gradings, followed by the K-12 teachers. Unlike the query-based summariser, a significant difference between the group giving the highest gradings and the other groups can only be shown for two groups: Computer Science and Arabic Literature students. As before, the lowest gradings came from the Computer Science students.

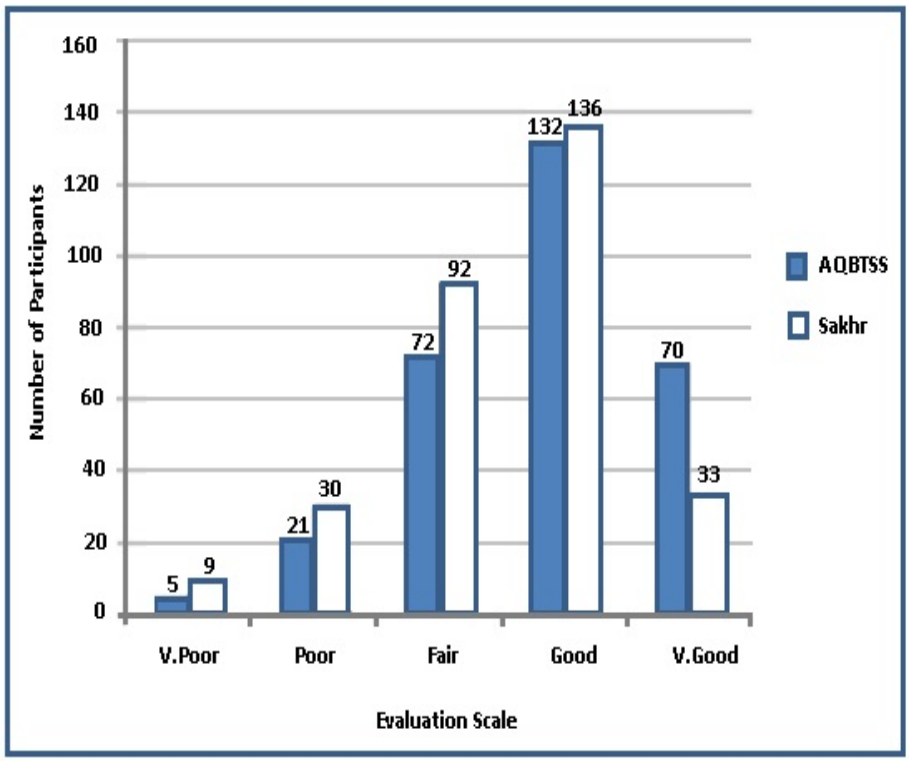

Figure 6.1: AQBTSS vs Sakhr

If we analyse each user group separately, we find that only the Humanities students did not show a significant difference in preference over one or the other summariser, although the average rating for the query-based summary (2.98) was also higher than for the concept-based summariser (2.97), see Table 6.5. All other groups appear to strongly prefer summaries generated by the query-based summariser. This preference is perhaps not surprising given that the summary is created for a specific query by AQBTSS, rather than a general set of conceptual terms (representing the particular category under which this document was classified) in the case of ACBTSS.

When comparing our query-based summariser AQBTSS with the Sakhr summariser we found that the "most critical" user group, i.e. the one that gave our summariser the lowest average with a score of 2.81, considered the commercial summariser to have a significantly worse performance, with an average score of 2.52. We hypothesise that a new experiment with any of the other user groups is likely to result in the

same preference. The most remarkable observation is that our query-based summariser resulted in a significantly higher average rating by the subjects than the rating for the commercial baseline.

The overall conclusion we draw from these experiments is that the query-based summariser performs much better than the concept-based summariser, and that it even outperforms a sensible baseline, as represented by Sakhr.

6.2 Automatic Evaluation

The automatic evaluation methods applied in this section are used to measure the quality of the summaries generated by AQBTSS, ACBTSS, *Gen–Summ*, *LSA–Summ*, Sakhr and Baseline–1 summarisers (Section 5.1). We used different evaluation methods to compare the summarisers from different points of view. The automatic evaluation results will determine which summarisation method, query-based or generic, will be used in the multi-document summarisation case.

The automatic evaluation metrics used to evaluate our single-document summarisers are ROUGE [Lin, 2004], AutoSummENG [Giannakopoulos et al., 2008] and Dice's Coefficient [Dice, 1945; Manning and Schütze, 1999]. ROUGE was the main evaluation metric in conferences such as the Text Analysis Conference (TAC) and the Document Understanding Conference (DUC) series. The reason behind using ROUGE measure was based on its suitability for evaluating extractive summaries as it counts the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary and the reference (gold–standard) summaries [Lin, 2004]. AutoSummENG was introduced in TAC-2011 MultiLing Summarisation Pilot[1], see Section 2.2 in Chapter 2.

The gold-standard (model) summaries provided to ROUGE and AutoSummENG

[1] `http://users.iit.demokritos.gr/~ggianna/TAC2011/MultiLing2011.html`

are the gold-standard summaries created for EASC corpus using different aggregation methods (*Level 2*, *Level 3* and *ALL*) explained earlier, see Section 4.1.2.

Dice's coefficient was used to judge the similarity of the sentence selections in EASC gold-standard summaries with those generated by Sakhr, AQBTSS, *Gen–Summ*, *LSA–Summ* and Baseline–1. Statistically significant differences can be observed in a number of cases, but we will concentrate on some more general observations.

6.2.1 Automatic Evaluation Results

In this section we show the results of the automatic evaluation methods used to evaluate our summarisers. These include ROUGE, AutoSummENG and Dice's Coefficient.

To measure how close the selection process of the summarisers is to the selection process done by the participants we used Dice's coefficient, the higher the percentage, the more close the selection process is.

For sets X and Y of keywords used in Information Retrieval (IR), Dice's coefficient may be defined as twice the shared information (intersection) over the combined set (union):

$$S = \frac{2|X \cap Y|}{|X| + |Y|}$$

Table 6.6 illustrates the results of this evaluation. We see that the commercial summariser Sakhr as well as the *Gen–Summ* and *LSA–Summ* most closely approximate the gold-standards (*Level 2* and *Level 3*). This is perhaps not surprising as the overlap with the document's first sentence has been shown to be a significant feature in many summarisers [Fattah and Ren, 2008; Yeh et al., 2008].

It is interesting to note that summaries consisting of a single sentence only (i.e. Baseline–1) do not score particularly well. That suggests that the first sentence is important but not sufficient for a good summary. When comparing Baseline–1 with

the *Level 2* and *Level 3* summaries, respectively, we also note how the "wisdom of the crowd" seems to converge on the first sentence as a core part of the summary.

LSA–Summ is the summariser that most closely approximates the *Level 2* gold-standard summaries. This is perhaps not surprising as LSA has been shown to work effectively in various NLP and IR tasks [Li et al., 2006].

Level	**Sakhr**	**AQBTSS**	**Gen–Summ**	**LSA–Summ**	**Baseline–1**
All	39.07%	32.80%	39.51%	39.23%	25.34%
Level 2	48.49%	39.90%	48.95%	50.09%	26.84%
Level 3	43.40%	38.86%	43.39%	42.67%	40.86%

Table 6.6: Dice's Results: Compare participants and summarisers selections

We also evaluated the summarisers using Dice's coefficient. The results are given in Table 6.7. The results suggest that the summaries of the Baseline–1 summariser, which extracts the first sentence, do not correlate well with those of the other systems. In contrast, *Gen–Summ* and *LSA–Summ* generate summaries that are highly correlated. This is consistent with results obtained when comparing each of these systems against the gold-standards (see Table 6.6). It also demonstrates that the difference between a standard vector space approach and LSA is not great for the relatively short documents in a collection of limited size [Dumais et al., 1988].

Summariser	**Sakhr**	**AQBTSS**	**LSA–Summ**	**Gen–Summ**	**Baseline–1**
Sakhr	—	51.09%	58.77%	58.82%	38.11%
AQBTSS	—	—	54.61%	58.48%	47.86%
LSA–Summ	—	—	—	84.70%	34.66%
Gen–Summ	—	—	—	—	34.99%

Table 6.7: Dice's Results: comparing systems.

In addition to using Dice's coefficient, we also applied ROUGE and AutoSummENG. In our experiments with AutoSummENG we obtained values for "CharGraphValue" (see Section 2.2 in Chapter 2) in the range [0.516–0.586]. *Gen–Summ* and *LSA–Summ* gave the highest values, indicating that they produce summaries more

similar to our gold-standard summaries than those of Sakhr and AQBTSS. When applying ROUGE we considered the results of ROUGE-2 (n-gram = 2) as it has been shown to work well in single-document summarisation tasks [Lin, 2004].

In order to measure the quality of the presented summarisers' extraction-process we used ROUGE-2 to find the recall and precision measurements as a step towards measuring the retrieving process of the relevant extracted sentences. This was done with regard to all the three levels: *All*, *Level 2* and *Level 3*, see Section 6.2. Table 6.8 shows the ROUGE-2 scores with the recall and precision of the Sakhr, AQBTSS, *Gen–Summ*, *LSA–Summ* and Baseline–1 summarisers against the derived gold-standards considering the *All* level. Table 6.9 shows the ROUGE-2, recall and precision results of those summarisers against the *Level 2* and *Level 3* derived gold-standards. The results displayed in the tables are with the absence of the Arabic light stemmer.

Observing the ROUGE-2 scores in Tables 6.8 and 6.9 we see that *LSA–Summ* and *Gen–Summ* performed better on average than the other summarisers when using *All*, *Level 2* and *Level 3* gold-standards. The results show that using the first sentence to summarise documents in *Gen–Summ* and *LSA–Summ* has slightly improved the summaries quality when compared to Sakhr and AQBTSS. This can be explained by observing the recall and precision scores of Baseline–1 summariser, which extracts the first sentence as a summary. Baseline–1's high precision scores strengthen the fact that the first sentence tends to contain important information about the content of the entire article, see Section 5.1.2. Baseline–1's low recall scores suggest that there have been other relevant sentences that have been overlooked. Regarding the other systems, they all performed better than Baseline–1. It is worth noting that the results obtained from ROUGE are in line with results from Dice's Coefficient and AutoSummENG.

To show the effect of using the Arabic light stemmer on retrieving extracted sentences we ran the same experiments again, but this time considering only our *Gen–Summ* and *LSA–Summ* summarisers. The reason is to show the effect of language–

dependent tools on the summaries quality. Calculating recall and precision, considering all sentences identified by any of the annotators (*All*), we found that the use of an Arabic light stemmer led to a slight improvement in both of the measurements, having the recall, precision and ROUGE-2 scores of the *Gen–Summ* summariser 53.63%, 40.57% and 46.20% respectively, and 53.94%, 39.82% and 45.82% for the *LSA–Summ* summariser. Considering *Level 2* and *Level 3* we also find a slight improvement to the results, which reconfirms that using natural language processing tools can have a positive impact when processing Arabic texts [Croft et al., 2009].

The results illustrated in this section showed that the generic summarisers (*Gen–Summ* and *LSA–Summ*) outperformed the other single-document summarisation methods (AQBTSS, ACBTSS and Sakhr). Based on this, we will continue using *Gen–Summ* and *LSA–Summ* (the generic summarisers that use the first sentence to summarise documents) when performing the experiments on multi-document summarisation. The summarisers will be modified to deal with multi-documents and to eliminate redundant sentences. The use of *Gen–Summ* and *LSA–Summ* in multi-document summarisation will be explained in detail in Section 5.2.1.

Measure	**Sakhr**	**AQBTSS**	**LSA–Summ**	**Gen–Summ**	**Baseline–1**
Recall	44.62%	35.23%	51.81%	50.42%	16.77%
Precision	42.09%	43.70%	39.35%	40.12%	66.41%
ROUGE-2	43.32%	39.01%	44.73%	44.68%	26.78%

Table 6.8: ROUGE-2: Recall/Precision results: comparing systems, “all levels” (no stemmer).

Level	Sakhr	AQBTSS	LSA–Summ	Gen–Summ	Baseline–1
Level 2 Recall	43.39%	34.96%	51.48%	49.11%	16.06%
Level 2 Precision	64.87%	66.44%	61.66%	60.88%	99.35%
Level 2 ROUGE-2	52.00%	45.81%	56.11%	54.36%	27.64%
Level 3 Recall	53.52%	44.51%	60.49%	59.85%	28.96%
Level 3 Precision	43.32%	49.27%	41.71%	42.36%	88.24%
Level 3 ROUGE-2	47.88%	46.77%	49.38%	49.61%	43.61%

Table 6.9: ROUGE-2: Recall/Precision results: comparing systems, "levels 2 and 3" (no stemmer).

6.3 The Effect of Summary Length on the Evaluation Scores

One argument that could be raised is that a major confounding factor in assessing summaries with ROUGE could simply be the length of the summary. In all the experiments conducted on single-document summarisation the gold-standard summaries' length varied. Longer summaries are statistically more likely to have more in common (overlapped sentences) with other summaries, than short summaries [Singhal et al., 1996]. At the time of writing, we were not successful in finding significant comparisons between results of short versus long summaries. In conferences such as the TAC-2011 MultiLing, summaries that are out-of-limit are penalised using a certain measure such as the Length-Aware Grading measure (LAG) (see Section 5.2.2). This could solve the problem of out-of-limit summaries, however it did not provide a definite answer on whether ROUGE scores increase when evaluating short or long summaries.

To address this point we ran a number of experiments using the EASC corpus gold-standard summaries (see Section 4.1.2). We only examined the gold-standard summaries as the length of the generated summaries is fixed by the summarisers' guidelines (see Section 5.1.2).

6.3.1 Experimental Setup

For this experiment we used the same summarisers examined in Section 6.2, those are Sakhr, AQBTSS, *LSA–Summ*, *Gen–Summ* and Baseline–1. *LSA–Summ* and *Gen–Summ* were used once with, and once without, an Arabic light stemmer. We calculated ROUGE score for each of the generated summaries against each one of the gold-standard summaries (each summary was evaluated against five gold-standard summaries).

We used the Spearman's correlation coefficient to judge the likely impact of changes of the gold-standard summaries length and ROUGE scores. Spearman's coefficient was used as it is the standard measure for ranked datasets [Myers and Well, 2003].

First, we looked for correlations between ROUGE scores against all the gold-standard summaries (765 summaries). Second, we considered the grouping of the gold-standard summaries (153 summaries each) of *Level 2*, *Level 3* and *ALL* (see Section 4.1.2).

6.3.2 Evaluation Results

Table 6.10 illustrates Spearman's correlation coefficient scores between the gold-standard summaries length and ROUGE scores. The gold-standard summary length varies between the levels. *No Grouping* means that all the gold-standard summaries have been considered. *No Grouping* summary length ranged between five words (one sentence) and 515 words (22 sentences) with an average of 114 words (five sentences) per summary. The *All* levels summary length ranged between 61 words (two sentences) and 781 words (39 sentences) with an average of 250 words (12 sentences) per summary. *Level 2* summary length ranged between 53 words (two sentences) and 570 words (26 sentences) with an average of 175 words (8 sentences) per summary. *Level 3* summary length ranged between 11 words (one sentence) and 296 words (17 sentences) with an

average of 98 words (4 sentences) per summary.

The Spearman scores in Table 6.10 show that ROUGE scores tend to increase when having shorter summaries. Taking the *LSA-Summ + Stemmer* summariser as an example, we can see that the correlation coefficient score is increasing when going up in levels. The highest correlation scores appear in *Level 3*, which contains the shortest summaries with an average of 98 words. This might suggest that ROUGE is sensitive to the length of the gold-standard summaries.

It should be noted that the grouping (aggregation) of summaries was meant to provide better gold-standards. The correlation scores justify the construction of the three-level gold-standard summaries (see Section 4.1.2 in Chapter 4).

For the multi-document summarisation experiments (Chapter 7), we fixed the summaries and gold-standards length (e.g. 240-250 words). This is to ensure consistency in ROUGE scores and to avoid any influence that may be caused by the length of the summaries.

Summariser	**No Grouping**	**All Levels**	**Level 2**	**Level 3**
Baseline–1	-0.1398210	-0.3662421	-0.5621326	-0.3906996
AQBTSS	-0.1538253	-0.0375638	-0.1396653	-0.0448962
Sakhr	-0.1965647	-0.1775809	-0.1344484	0.1168250
Gen-Summ	-0.1621263	0.1032594	0.1434433	0.2498756
LSA-Summ	-0.2061436	0.1623897	0.2096083	0.2549235
Gen-Sum + Stemmer	-0.1808086	0.0581310	0.1726065	0.3008074
LSA-Summ + Stemmer	-0.2097866	0.1623897	0.2387514	0.3181844

Table 6.10: Spearman's Correlation: ROUGE vs Summary Length

6.4 Summary

The comparisons between query-based, concept-based and generic summarisers using different gold-standard summaries acted as the backbone for the multi-document summarisation experiments. The comparison suggested that generic single-document sum-

marisers are capable of generating high quality summaries by using the first sentence to identify the topic to be summarised. We believe that the results of the different evaluation methods could help researchers in advancing the work on Arabic single-document summarisation. The results also showed the effect of using natural language processing techniques on the quality of the summarisation process. The work illustrated in this Chapter was published in El-Haj et al. [2009, 2010].

Chapter 7

Multi-document Summarisation Evaluation

This chapter shows the evaluation results of the work done on multi-document summarisation presented in Chapter 5. Both human generated and automatically generated corpora were used in the evaluation. The evaluation results of the multi-document summarisation experiments and redundancy elimination techniques are presented and compared with the results of other summarisation techniques and systems in the literature.

The Chapter is divided into four main sections. Each section shows the evaluation of one of the multi-document summarisation approaches presented in Section 5.2. Section 7.1 presents the evaluation results of the statistical-based and semantic-based multi-document summarisation approaches. The section also shows the comparison between the different redundancy elimination techniques, which includes using Vector Space Model, Latent Semantic Analysis and Dice's Coefficient. Section 7.2 shows the evaluation of the various cluster-based summarisation approaches used. The section also presents the evaluation results of comparing the cluster-based redundancy elimination techniques with the redundancy elimination techniques in Section 7.1. Section

7.3 shows the evaluation results of our participation at the TAC–2011 MultiLing Summarisation workshop. Section 7.4 presents the evaluation and comparison results of the experiments done on multi-document summarisation with the use of NLP tools, as described in Section 5.2.3.

7.1 Statistical and Semantic Summarisation Evaluation Results

In this section we show the results of our statistical and semantic summarisers presented in Section 5.2.1. To see the quality of these summarisers we compared the ROUGE scores of our summarises with those published by Mihalcea and Tarau [2005]. In their work they published the ROUGE results for the top five systems participated in DUC–2002 multi-document summarisation task. The systems used the DUC–2002 dataset, the same one we are using, which makes the comparison between our work and theirs feasible. The ROUGE results published by Mihalcea and Tarau [2005] were conducted in 2005 using an earlier version of ROUGE. As the metric has changed we decided to replicate the baseline experiment performed in their work. For each reference set in the DUC–2002 dataset their baseline summariser selects the first sentence of each article and combines them all as a multi-document summary, which resulted in a number of 59 summaries of the 59 reference sets available.

Table 7.1 illustrates the results of our Arabic and English multi-document summarisers. In the table we refer to the Arabic and English summarisers with the suffix *Arb* and *Eng* respectively. Dice, VSM and LSA stands for the redundancy elimination techniques used. As illustrated in the table both our Arabic and English summarisers using the Dice's coefficient redundancy elimination technique performed better on average than using the VSM and LSA models. Observing the recall and precision scores

in the same table, we found that the results are close. In order to demonstrate that these results are sound we compared the recall and precision of each summary individually. Having the recall and precision for of each summary (see Figures 7.1, 7.2 and 7.3) we found that the recall and precision do vary. What we see in the table is the overall recall and precision which equals the average instance recall and precision [Nielsen, 2008].

System	ROUGE-1	Precision	Recall
Dice–Sum–Arb	0.37085	0.37366	0.36915
Dice–Sum–Eng	0.36945	0.36950	0.36958
VSM–Sum–Arb	0.36411	0.39115	0.34784
VSM–Sum–Eng	0.35666	0.37998	0.34220
Our Baseline	0.29457	0.29505	0.29421
LSA–Sum–Eng	0.28986	0.39071	0.23810
LSA–Sum–Arb	0.28734	0.41296	0.22986

Table 7.1: Summarisation with Redundancy Elimination (no Clustering)

Although the results of our Arabic and English summarisers using LSA redundancy elimination technique did not perform better than the baselines, they gave good precision scores. As illustrated in Table 7.1, the precision of both summarisers was relatively high compared to when using the other redundancy elimination models. This lends support to the idea that using LSA increases similarity between sentences and thus increases precision [Dumais et al., 1988].

System	ROUGE-1
System26	0.3578
System19	0.3450
System28	0.3435
System29	0.3264
System25	0.3056

Table 7.2: Top 5 Systems in DUC 2002.

We found that our summarisers' results are comparable to the ROUGE-1 results of the top five summarisers in the DUC–2002 competition (see Table 7.2).

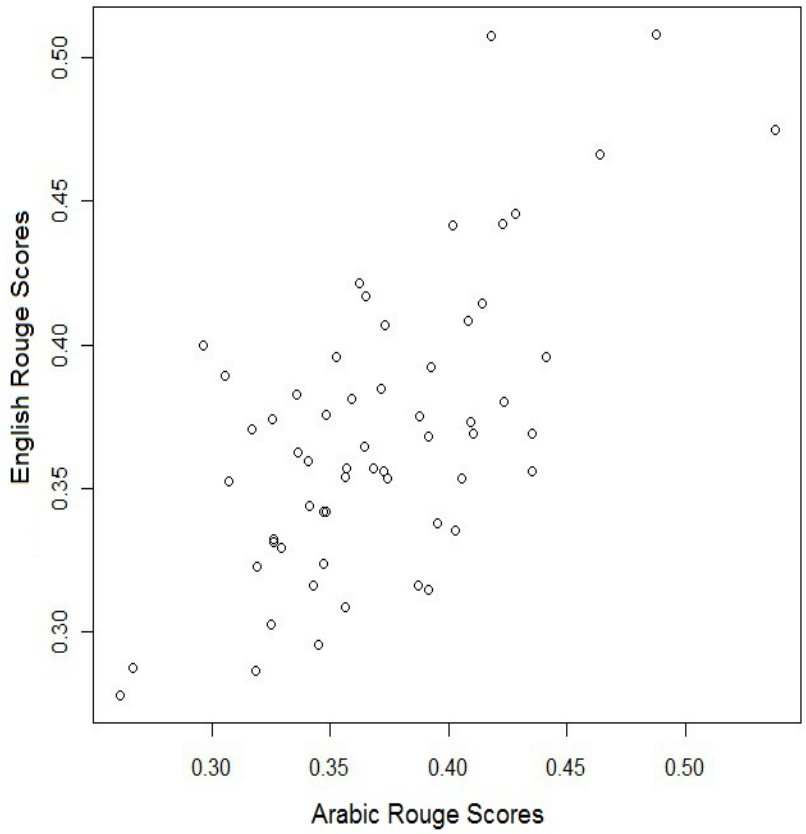

Figure 7.1: Dice ROUGE-1 Score Distribution

System	Observations	p
LSA–Sum–Eng vs LSA–Sum–Arb	59	0.80748
Dice–Sum–Eng vs Dice–Sum–Arb	59	0.79683
VSM–Sum–Eng vs VSM–Sum–Arb	59	0.24888

Table 7.3: t.test results (Not Significant at $p<0.05$).

To determine significance we performed standard t.tests ($p<0.05$) using R Project for Statistical Computing[1]. The t.test is performed by testing the ROUGE-1 score of each summary generated by the English and Arabic summarisers. The results of significance testing (Table 7.3) showed no significant difference between the English and the Arabic summarisers. This shows that our English and Arabic summarisers are performing on par with each other.

We ran another t.test to test whether our English and Arabic systems perform

[1]http://www.r-project.org/

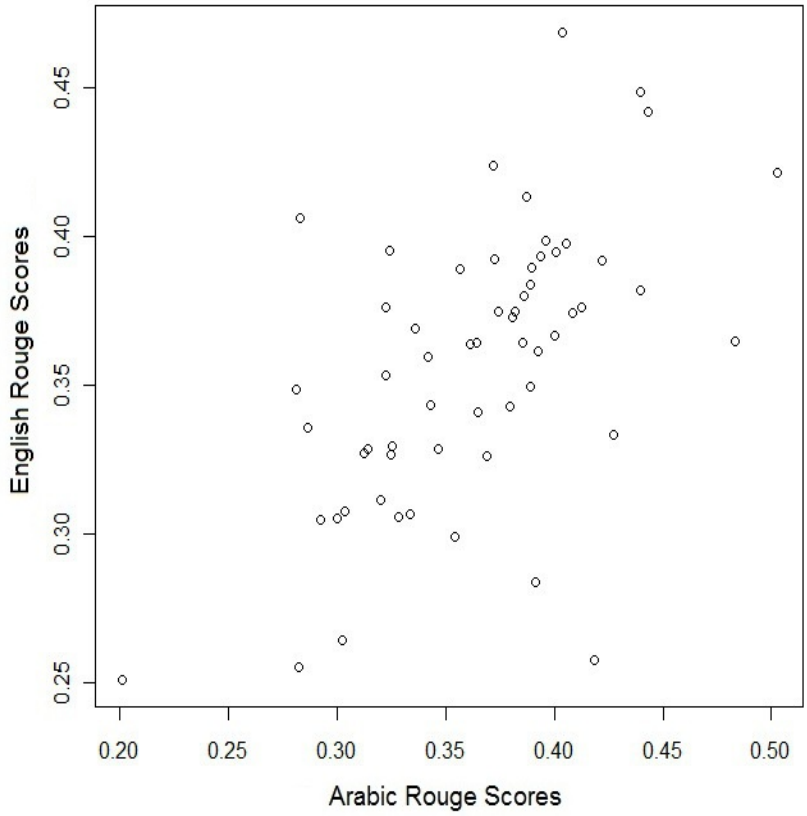

Figure 7.2: VSM ROUGE-1 Score Distribution

System	Observations	p
Dice–Sum–Arb vs Baseline	59	1.80E-13
Dice–Sum–Eng vs Baseline	59	6.39E-14
VSM–Sum–Arb vs Baseline	59	1.53E-13
VSM–Sum–Eng vs Baseline	59	1.92E-13

Table 7.4: t.test Systems vs Baseline results (Significant at $p<0.05$).

significantly better than the baseline. Table 7.4 shows that our systems perform significantly better than the baseline ($p<0.05$), although note that Table 7.1 shows that the Arabic and English LSA-Summariser do not perform as well as the baseline.

To compare the Arabic and English summarisers with each other we measured the selection agreements between the two sets of summaries using Dice's coefficient, recall/precision and F-measure. Table 7.5 illustrates the results of this comparison. The high precision percentage in the same table shows the accuracy of the translation

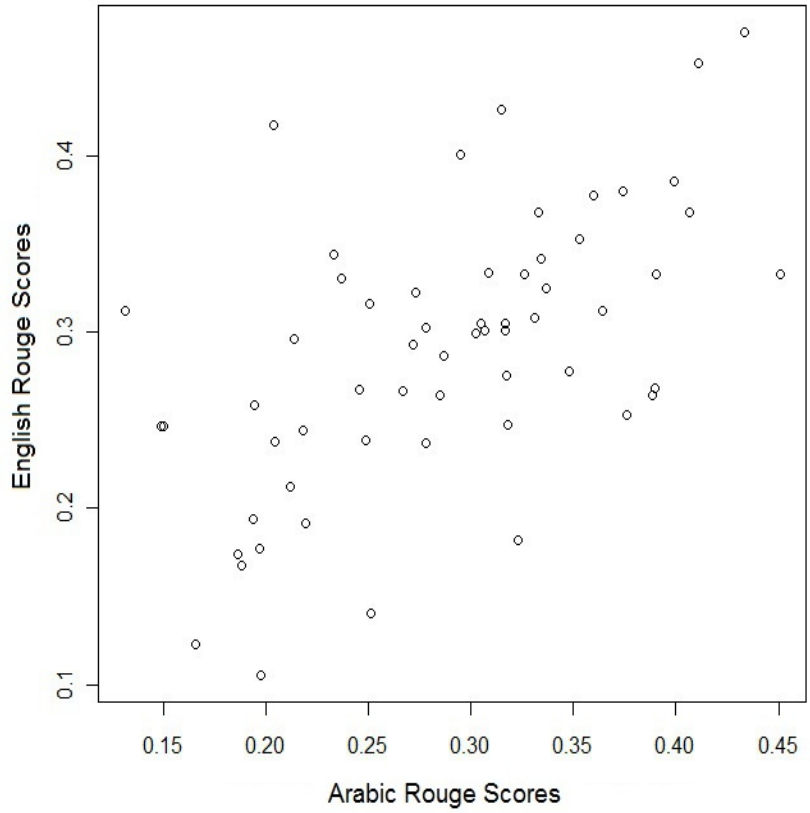

Figure 7.3: LSA ROUGE-1 Score Distribution

system with 80% of the extracted sentences considered relevant. This means that the translated sentences still hold the original meaning when translated from English into Arabic.

Metric	**English vs Arabic DUC–2002**
Dice's	66.27%
Recall	62.49%
Precision	80.50%
F Measure	70.36%

Table 7.5: English vs Arabic (DUC–2002) Selection Agreement

One of our main findings of this experiment is that automatic machine translation of an English dataset into Arabic is a viable and economic alternative to the manual creation of an Arabic dataset and gold-standard summaries. It also permits the performance of Arabic summarisers to be compared with that of English summarisers

(Table 7.2). We believe that this finding will help the field of Arabic automatic multi-document text summarisation, as it points to a cost-effective solution to the problem of limited Arabic resources. The work presented in this section has already been published [El-Haj et al., 2011a].

7.2 Cluster-based Summarisation Evaluation Results

In this section we show the evaluation results of the cluster-based summarisation experiments presented in Section 5.2.2.

Tables 7.6 and 7.7 respectively illustrate the results of our Arabic summarisers applying K-means clustering to *all* sentences. The highest ROUGE scores are displayed in bold. Similarly, Tables 7.8 and 7.9 represent the results of clustering for redundancy elimination.

Number of Clusters	ROUGE-1	Precision	Recall
1	**0.3899**	0.3849	0.3952
2	0.3701	0.3660	0.3744
5	0.3736	0.3697	0.3778
10	0.3848	0.3800	0.3899
15	0.3882	0.3839	0.3930
20	0.3898	0.3853	0.3946

Table 7.6: Clustering all Sentences (Biggest Cluster)

Number of Clusters	ROUGE-1	Precision	Recall
1	0.1725	0.6733	0.0997
2	0.2322	0.5495	0.1496
5	0.3152	0.4516	0.2469
10	0.3822	0.3954	0.3729
15	**0.3878**	0.3868	0.3898
20	0.3848	0.3819	0.3879

Table 7.7: Clustering all Sentences (First Sentence from each Cluster)

We draw the following conclusions:

Number of Clusters	ROUGE-1	Precision	Recall
1	0.3806	0.3761	0.3852
2	**0.3817**	0.3788	0.3853
5	0.3591	0.3794	0.3504
10	0.3346	0.3991	0.3134
15	0.2718	0.4270	0.2290
20	0.1704	0.4614	0.1149

Table 7.8: Redundancy Elimination (Biggest Cluster)

Number of Clusters	ROUGE-1	Precision	Recall
1	0.1421	0.4680	0.0848
2	0.2279	0.4267	0.1570
5	0.3638	0.3848	0.3470
10	0.3645	0.3616	0.3676
15	0.3676	0.3638	0.3717
20	**0.3820**	0.3777	0.3865

Table 7.9: Redundancy Elimination (First Sentence from each Cluster)

1. The best overall ROUGE score is obtained by clustering *all* sentences and creating a single "cluster", i.e. by selecting sentences to form the summary that are most similar to the centroid of all sentences. In fact, when selecting the biggest cluster of *all* sentences we observe that the number of clusters in our experiments does not have a significant effect on the ROUGE scores. All results are fairly consistent with no significant difference measurable between any pair of results.

2. An alternative way to obtain top ROUGE scores is to use clustering for redundancy elimination (as illustrated in Tables 7.8 and 7.9). However, in this case the experimental settings do have a significant impact. If the number of clusters is small (1 or 2) when clustering a pre-selected set of sentences and then using the biggest cluster to form the summary, we get a ROUGE score that is significantly better than using number of clusters of 10 or above. Conversely, we get a ROUGE value marginally smaller than the top overall score by selecting the centroids from a large set of clusters. Given our approach to eliminate redundancy

using clustering we would in fact expect the ROUGE scores (as well as precision and recall) to vary across different number of clusters and selection methods. As this suggest different summaries of different sizes to be generated.

3. We also observe that for certain experimental settings the cluster-based redundancy elimination approach can (marginally) improve our previously applied redundancy elimination steps which are using no clustering (Table 7.1).

4. Throughout the experiments we observe that high precision goes hand in hand with relatively low ROUGE score, i.e. low recall values.

5. We also found that *none* of the pairwise comparisons of ROUGE scores, when comparing the Arabic summariser with the corresponding English summariser are significant. This is an indication that the automatic translation process did not affect the summarisation quality.

The ROUGE-1 results of the five best performing summarisers in DUC–2002 are given by Mihalcea and Tarau [2005]. The results are reproduced in Table 7.2. Our multi-document summarisers achieve slightly higher ROUGE-1 scores than the top systems reported at DUC–2002. This is true for the English system (top ROUGE score 0.3856), but more importantly for this work, also for our Arabic multi-document summariser (working on the Arabic translation of the dataset).

The main finding of our experiments appears to be the fact that a simple centroid-based similarity clustering with a single "cluster" when performing summarisation could be considered an alternative way to the use of different cluster numbers. The work by Radev et al. [2000, 2004]; Sarkar [2009] considered a variable number of clusters, whereas our experiments demonstrate that for the given test collection the closeness to the centroid (to identify the important sentences) can produce summaries with similar quality. The work on cluster-based summarisation presented in this section has already been published [El-Haj et al., 2011b].

7.3 TAC–2011 MultiLing Summarisation Evaluation Results

This subsection shows the evaluation results of our participation at the TAC–2011 MultiLing Summarisation workshop. It also shows our performance compared to other state-of-the-art summarisers for both Arabic and English languages.

Apart from the actual participants in the MultiLing task there was also a global *baseline* (System ID9) and a global *topline* (System ID10), see Table 7.10. The global baseline determines the centroid of the document set in the space. Given the centroid it finds the text that is most similar to the centroid and uses it in the summary. The global baseline will keep adding text to the summary until it reaches a summary limit. If no summary limit is given the baseline could select the whole text in the summarised articles as a summary. The system does not use any sentence selection technique. The global topline uses the human model summaries instead of the text collection (thus cheating). Then produces random summaries by combining sentences from the original texts with the given human summaries.

For the Arabic language, there were seven participants (peers) in addition to the two baseline systems, for a total of nine runs. The English language had eight participants in addition to the two baseline systems, for a total of ten runs. Our system in both the Arabic and the English competition is referred to as **ID8**. According to MultiLing guidelines and to avoid inconsistency in ROUGE scores, summaries that are out-of-limit were penalised using the Length-Aware Grading measure (LAG) as in Equation 7.1:

$$LAG(g, S) = g * \big(1 - \frac{\max(\max(l_{min} - |S|, |S| - l_{max}), 0)}{l_{min}}\big) \tag{7.1}$$

Figures 7.4 and 7.5 illustrate the overall responsiveness (how well is the sum-

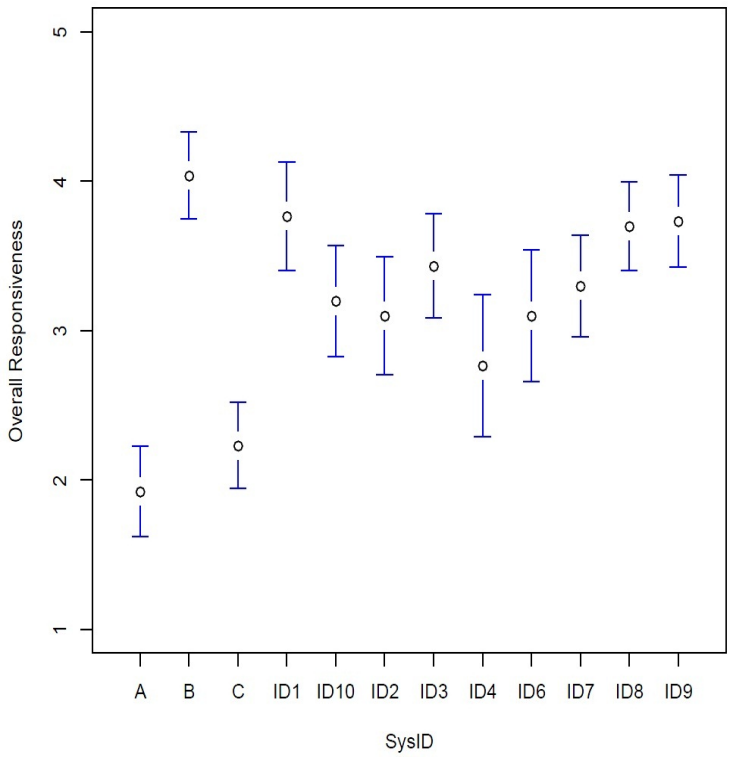

Figure 7.4: Arabic Overall Responsiveness — All Peers

SysID	Human (Overall)	Human (LAG)
ID1	3.77	3.77
ID9	3.73	3.73
ID8	3.70	3.66
ID3	3.43	3.30
ID7	3.30	3.20
ID10	3.20	3.10
ID2	3.10	3.10
ID6	3.10	2.76
ID4	2.77	2.76

Table 7.10: Arabic Overall and LAG Responsiveness Scores

mary responding to the information need contained in the topic statement) [Dang and Owczarzak, 2008] and LAG of all the systems of the Arabic language including the human peers. On the other hand Figures 7.6 and 7.7 show the overall responsiveness and LAG of all the systems of the English language including the human peers. A, B and C are the human peers, see Section 4.2.2.

SysID	Human (Overall)	Human (LAG)
ID3	3.83	3.55
ID2	3.53	3.53
ID10	3.20	3.20
ID1	3.20	3.10
ID5	3.03	2.92
ID8	2.73	2.73
ID9	2.50	2.50
ID7	2.30	2.29
ID6	2.67	2.20
ID4	2.033	2.033

Table 7.11: English Overall and LAG Responsiveness Scores

SysID	Recall	Precision	F-Measure
ID10	0.46751	0.25828	0.30786
ID3	0.37218	0.29644	0.29987
ID2	0.34194	0.29444	0.29188
ID6	0.35648	0.25396	0.2763
ID8	0.38854	0.22008	0.26786
ID4	0.42259	0.20676	0.26279
ID1	0.29869	0.21359	0.2319
ID9	0.32405	0.23596	0.23097
ID7	0.24058	0.22703	0.22376

Table 7.12: Arabic ROUGE-1 Scores

SysID	Recall	Precision	F-Measure
ID10	0.23394	0.13669	0.14922
ID3	0.15808	0.13857	0.1278
ID6	0.13767	0.0992	0.10629
ID2	0.13858	0.09774	0.10347
ID8	0.14726	0.07851	0.09653
ID9	0.12559	0.10451	0.09497
ID1	0.12057	0.08482	0.0889
ID4	0.13962	0.06886	0.08634
ID7	0.10627	0.07772	0.08577

Table 7.13: Arabic ROUGE-2 Scores

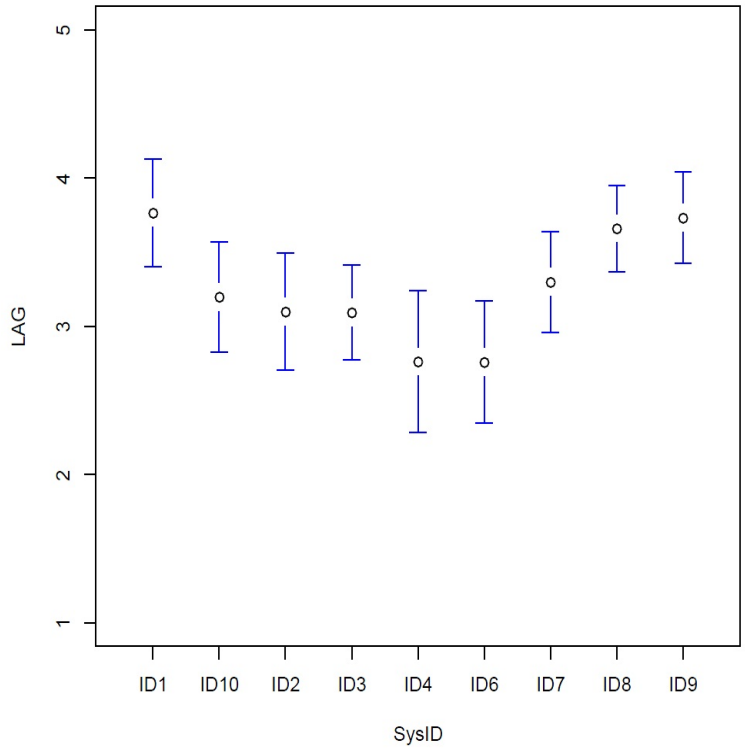

Figure 7.5: Arabic LAG for Overall Responsiveness — Systems Only

Tables 7.10 and 7.11 illustrate the overall and LAG measure for systems participating in the Arabic and English languages respectively. As we can see in the first table, our system (ID8) was judged by human assessors to be among the top systems. Only two systems (one of them the baseline) scored better, but not by a large margin. The LAG grade of our system reflect that some of our summaries were out of limit (below 240 words), but as we can see, this did not affect the ranking.

According to the results of the English human evaluation in Table 7.11, we see that our system performed better than the baseline. However, we note that the scores given by human assessors is substantially lower than for the Arabic system.

Tables 7.12, 7.13 and 7.14 illustrate the ROUGE results and the ranking of our Arabic multi-document summariser (System ID8) as well as the corresponding AutoSummENG-MeMoG evaluation metric in Table 7.18. The ROUGE results correlate quite closely with the AutoSummENG–MeMoG ranking of systems.

SysID	Recall	Precision	F-Measure
ID10	0.2783	0.14152	0.15489
ID3	0.19889	0.16758	0.1514
ID2	0.16618	0.14293	0.13309
ID6	0.17617	0.1145	0.12456
ID8	0.18475	0.09219	0.11487
ID4	0.20836	0.07856	0.1071
ID7	0.11818	0.09413	0.09874
ID1	0.14033	0.09419	0.09871
ID9	0.15185	0.11618	0.0974

Table 7.14: Arabic ROUGE-SU4 Scores

SysID	Recall	Precision	F-Measure
ID10	0.52488	0.51806	0.52141
ID2	0.46481	0.45655	0.46062
ID3	0.43169	0.47909	0.45404
ID4	0.44423	0.44966	0.44691
ID5	0.41092	0.43513	0.42243
ID1	0.40524	0.41253	0.40776
ID6	0.3547	0.45122	0.39617
ID7	0.39586	0.3953	0.39547
ID8	0.38714	0.39265	0.38985
ID9	0.38105	0.37726	0.3791

Table 7.15: English ROUGE-1 Scores

SysID	Recall	Precision	F-Measure
ID10	0.25177	0.2483	0.25
ID3	0.1733	0.19256	0.18237
ID2	0.17052	0.16779	0.16914
ID4	0.1517	0.15369	0.15269
ID5	0.13605	0.14404	0.13985
ID1	0.12125	0.12448	0.12247
ID8	0.12144	0.12298	0.12219
ID6	0.10655	0.1367	0.11937
ID9	0.10962	0.10841	0.109
ID7	0.09662	0.09612	0.09635

Table 7.16: English ROUGE-2 Scores

SysID	Recall	Precision	F-Measure
ID10	0.27248	0.26882	0.27062
ID1	0.15995	0.16322	0.16112
ID2	0.2022	0.19868	0.20042
ID3	0.19927	0.22148	0.20973
ID4	0.19083	0.1932	0.192
ID5	0.17475	0.18503	0.17964
ID6	0.1457	0.18648	0.16312
ID7	0.14507	0.1446	0.1448
ID8	0.1566	0.15874	0.15765
ID9	0.14805	0.14655	0.14728

Table 7.17: English ROUGE-SU4 Scores

SysID	MeMoG
ID10	0.665674
ID3	0.482755
ID4	0.382946
ID2	0.368587
ID6	0.340396
ID8	0.305233
ID1	0.296868
ID9	0.282094
ID7	0.261209

Table 7.18: Arabic AutoSummENG–MeMoG Scores

SysID	MeMoG
ID10	0.5477871
ID3	0.4256148
ID2	0.3859586
ID4	0.3785725
ID5	0.3500278
ID6	0.3490875
ID1	0.3443412
ID8	0.3323676
ID7	0.3108508
ID9	0.304319

Table 7.19: English AutoSummENG–MeMoG Scores

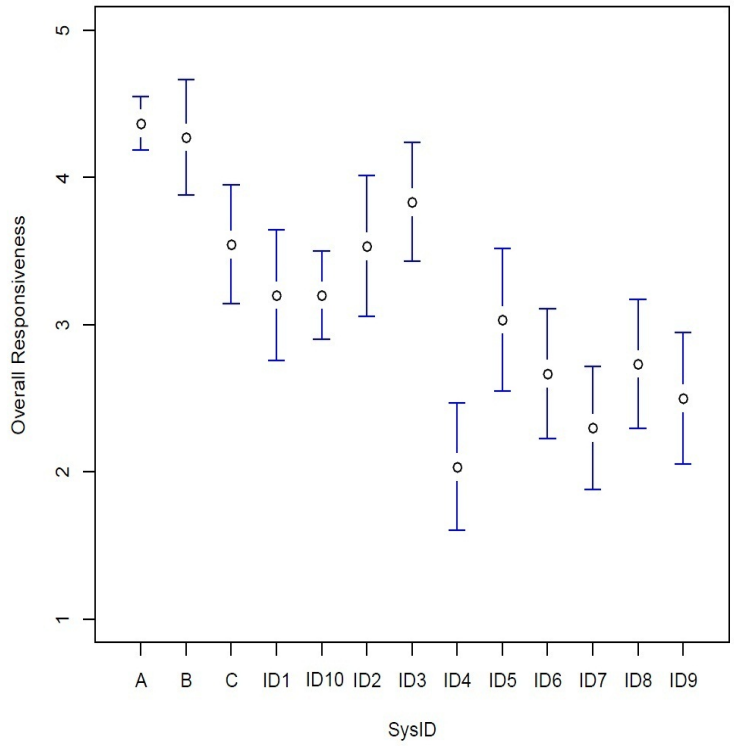

Figure 7.6: English Overall Responsiveness — All Peers

We observed that the automatic evaluation results place our Arabic summariser further down in the ranked lists of systems compared to the human assessment. This is an area for future work as this seems to suggest that the automatic evaluation metrics are not necessarily in line with human judgements.

Tables 7.15, 7.16 and 7.17 give the ROUGE results and the ranking of our English multi-document summariser, Table 7.19 has the AutoSummENG-MeMoG evaluation results. As with the Arabic summariser, we note that the human assessment places our system higher in the ranked order than the automatic evaluation. Details of our participation at TAC MultiLing workshop have already been published in El-Haj et al. [2011c]; Giannakopoulos et al. [2011].

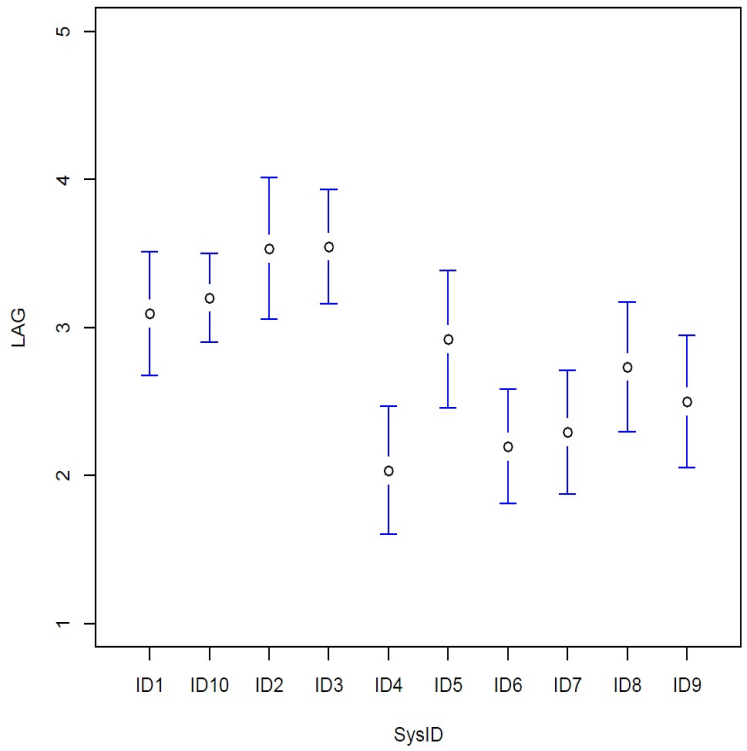

Figure 7.7: English LAG for Overall Responsiveness — Systems Only

7.4 Summarisation with NLP Tools Results

In this subsection we show the evaluation and comparison results of the experiments done on multi-document summarisation with the use of NLP tools, as described in Section 5.2.3. We will start by showing the results of the Arabic language experiments, followed by the results of the English language experiments. The section concludes by comparing the results of the two languages with the results we achieved at the TAC–2011 workshop (see Section 7.3).

Table 7.20 gives the results of our experiments with the summarisers that apply Arabic NLP tools to the summarisation process. As before, the evaluation was done using the ROUGE-1 evaluation metric [Lin, 2004]. The table displays the recall, precision and F-measure scores generated by ROUGE. The scores are calculated using the gold-standard summaries (human summaries) of the TAC MultiLing Arabic corpus.

We refer to recall, precision and F-measure with using 'R', 'P' and 'F' respectively.

Settings	R	P	F
Domain Specific Stop List Only	0.40166	0.28954	0.30697
Light Stemmer + Standard Stop List	0.37465	0.23409	0.27348
Root-Stemmer + Domain Specific List	0.37177	0.24452	0.26786
Light-Stemmer Only	0.30359	0.22339	0.24519
Root-Stemmer + Standard Stop List	0.40615	0.18322	0.23346
Light-Stemmer + Domain Specific Stop List	0.29271	0.19759	0.21211
Standard Stop List Only	0.34885	0.19073	0.20995
Root-Stemmer Only	0.27958	0.18519	0.20379
Domain Specific + Standard Stop Lists	0.11429	0.20758	0.12456

Table 7.20: Arabic with NLP tools ROUGE-1 Scores

Table 7.21 shows the results of the experiments where we apply English NLP tools. The English gold-standard summaries (human summaries) were used to run ROUGE and generate the recall, precision and F-measure scores.

Settings	R	P	F
Lovins Stemmer + Domain Specific Stop List	0.38204	0.38912	0.38543
Porter Stemmer + Domain Specific Stop List	0.37195	0.37865	0.37524
Domain Specific Stop List Only	0.37149	0.37811	0.37473
Lovins Stemmer + Standard Stop List	0.36215	0.36866	0.36533
Porter Stemmer Only	0.36104	0.36883	0.36488
Porter Stemmer + Standard Stop List	0.35941	0.36657	0.36292
Lovins Stemmer Only	0.35934	0.36294	0.36111
Domain Specific + Standard Lists	0.35177	0.35903	0.35534
Standard Stop List Only	0.35057	0.35649	0.35348

Table 7.21: English with NLP tools ROUGE-1 Scores

In order to show the effect of using NLP tools on the summarisation process for both Arabic and English languages we compared the evaluation results with our results for the TAC–2011 MultiLing workshop, see Tables 7.12 and 7.15.

Comparing the results of using Arabic NLP tools with the results in Table 7.12 we found that we get very close to the *topline* system (ID10), which indicates that using NLP tools on Arabic language does have a positive effect. To check our findings we ran

a pairwise *t.test* on our best performing system comparing it with the topline ID10, and we found no significant difference, which shows that simply using an Arabic domain specific stop list as part of our system setup achieves results that are on par with the best-performing TAC–2011 submission according to the ROUGE-1 F-measure. This reconfirms that using NLP tools is essential for Arabic [Croft et al., 2009]

Running a *t.test* for the English settings, comparing for significance between our system (language independent) and our "advanced" pipeline, we found that NLP tools do not have such a positive effect when applying them to English summarisation.

We also found that stop words (particularly domain-specific) and some stemming, have a positive effect and improved the quality of Arabic summaries when comparing the scores we achieved with ones in Table 7.20. In contrast, using stop word lists and stemming did not improve the summaries quality of the English language.

7.5 Summary

Human and automatic evaluation metrics have been used to evaluate our multi-document summarisers. Both language–dependent and language–independent approaches have been applied. Language–independent summarisation included the use of statistical models such as the vector space model and Dice's coefficient. The evaluation results of the Arabic multi-document summarisation showed a slight improvement on the summarisation approach when applying natural language processing tools. Applying human and automatic evaluations on our different summarisation techniques showed that our work, and the results we achieved, contribute to the field of automatic summarisation by providing resources and evaluation results that can be used to advance the research on multi-document summarisation.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In our work we addressed the issue of text summarisation with a particular focus on Arabic multi-document summarisation. This required advancements in at least two areas, first of all it required the creation of Arabic test collections. The second area concerned the actual summarisation process to find methods that improve the quality of Arabic summaries. We addressed both points.

First, we created single and multi-document Arabic datasets both automatically and manually using a commonly used English dataset and by having human participants. We have demonstrated how gold-standard summaries can be extracted using the "wisdom of the crowd". Using crowd-sourcing allowed us to produce a resource to evaluating Arabic single-document extractive summaries at relatively low cost, this resource is now available to the community. For the case of Arabic multi-document summarisation we addressed the problem of the shortage of readily available resources. We achieved this by translating articles from English into Arabic using machine translation technique. Based on our findings we would argue that the translation process did not seem to affect the summarisation quality pointing at a feasible alternative for

the creation of test collections in under-resourced languages. Our work also suggested that summarising articles translated into Arabic provide good quality summaries that are comparable to English summarisers.

Being part of the organising committee for the TAC-2011 MultiLing (multi-lingual) summarisation pilot, we created and provided Arabic resources for multi-document summarisation purposes, which included human and system generated summaries in addition to a human translated corpus. We believe these resources could provide a useful benchmark for those developing Arabic single and multi-document summarisation tools.

Second, we developed extractive language dependent and language independent single and multi-document summarisers, both for Arabic and English. In our work we provided state-of-the-art approaches for Arabic multi-document summarisation. We explored clustering for multi-document Arabic summarisation. We investigated how clustering can be applied to multi-document summarisation as well as for redundancy elimination within this process. We used different parameter settings including the cluster size and the selection model applied in the extractive summarisation process.

Using ROUGE, precision/recall and AutoSummENG metrics we were able to measure the effect of applying different tools and methods. One of our main findings is that selecting sentences similar to the centroid of all sentences in the collection of related documents gives the highest ROUGE scores. We also showed that the Arabic (as well as English) summarisation system we developed has comparable performance with the top performing (English summarisation) systems at DUC-2002 and TAC-2011 MultiLing summarisation Pilot.

The work in the thesis is eclectic as a result of trying to demonstrate that summarization in Arabic is possible, and can replicate what is done in other languages.

Researchers on Arabic multi-document summarisation now have resources, tools and results that can be used to advance the research on Arabic single and multi-

document summarisation.

8.2 Future Work

Among future work is the application of more fine-tuned clustering to improve results furthermore. Experimenting with more language-specific features, such as morphological parsers, textual entailment and anaphoric resolution is an open research for more improvements in the future.

At the time of writing, researchers on Arabic Natural Language Processing (NLP) were not yet successful in tackling the field of Arabic abstractive summarisation. Abstractive summarisation requires an understanding of the original text and regenerating it in a shortened version. This is different from extractive summarisation as it involves the use of Natural Language Generation (NLG) tools to paraphrase the corpus using novel sentences. The lack of Arabic resources and NLG tools made it hard for researchers to successfully tackle this field. This can be solved by building Arabic NLG tools and resources including Arabic lexicons/WordNet and language models to develop Arabic abstractive summarisers that can generate coherent sentences.

With more Arabic tweets available on news websites such as *BBC Arabic* and *Arabic CNN*, one can build a real-time summariser to generate abstractive summaries for ongoing events, the summariser can work incrementally, where the generated summary will be updated as long as the event is still going and more tweets are being produced.

There is a plenty of room for more research on Arabic automatic summarisation. Enhancing the current Arabic summarisation techniques and methods is highly dependent on advancing the work on Arabic NLP and the availability of more Arabic tools and resources.

References

S. Abuleil, K. Alsamara, and M. Evens. Acquisition System for Arabic Noun Morphology. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, SEMITIC'02, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

A. Aker, T. Cohn, and R. Gaizauskas. Multi-document summarization using A* search and discriminative training. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP'10, pages 482–491, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

A. Aker., M. El-Haj, U. Kruschwitz, and D. Albakour. Assessing Crowdsourcing Quality through Objective Tasks. In *8th Language Resources and Evaluation Conference*, Istanbul, Turkey, 2012. LREC 2012.

H. Al-Ameed, S. Al-Ketbi, A. Al-Kaabi, K. Al-Shebli, N. Al-Shamsi, N. Al-Nuaimi, and S. Al-Muhairi. Arabic Light stemmer: A new Enhanced Approach. In *The 2nd International Conference on Innovations in Information Technology*, Dubai, United Arab Emirates, 2006. IIT'05.

H. Al-Muhtaseb and C. Mellish. Towards an Arabic Upper Model: A proposal. In *Proceedings of the 15th National Conference of Computers*, Dhahran, Saudi Arabia, 1997. King Fahd University of Petroleum and Minerals.

E. Al-Shammari and J. Lin. Towards an error-free Arabic stemming. In F. Lazarinis, E. Efthimiadis, J. Vilares, and J. Tait, editors, *Proceeding of the 2nd ACM workshop on Improving Non English Web Searching, iNEWS 2008, Napa Valley, California, USA, October 30, 2008*, pages 9–16. ACM, 2008. ISBN 978-1-60558-416-4.

L. Al-Sulaiti, ES. Atwell, and E. Steven. The design of a corpus of Contemporary Arabic. *International Journal of Corpus Linguistics*, 11(2):135–171, 2006. ISSN 1384-6655.

M. Albakour, U. Kruschwitz, and S. Lucas. Sentence-Level Attachment Prediction. In Hamish Cunningham, Allan Hanbury, and Stefan Rger, editors, *Advances in Multidisciplinary Retrieval*, volume 6107 of *Lecture Notes in Computer Science*, pages 6–19. Springer Berlin / Heidelberg, 2010.

M. Alghamdi, M. Chafic, and M. Mohamed. Arabic Language Resources and Tools for Speech and Natural Language: KACST and Balamand . In *The 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2009.

R. Alguliev and R. Aliguliyev. Effective Summarization Method of Text Documents. In A. Skowron, R. Agrawal, M. Luck, T. Yamaguchi, P. Morizet-Mahoudeaux, J. Liu, and N. Zhong, editors, *International Conference on Web Intelligence (WI)*, pages 264–271, Compiegne, France, 2005. IEEE Computer Society. ISBN 0-7695-2415-X.

R. Aliguliyev. A Novel Partitioning-Based Clustering Method and Generic Document Summarization. In *IAT Workshops*, pages 626–629. IEEE Computer Society, 2006. ISBN 0-7695-2749-3.

J. Allan, R.Gupta, and V. Khandelwal. Temporal summaries of news topics. In B. Croft, D. Harper, D. Kraft, and J. Zobel, editors, *The 24th International Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana, USA, 2001. ACM.

J. Allan, J. Aslam, N. Belkin, C. Buckley, J. Callan, B. Croft, S. Dumais, N. Fuhr, D. Harman, D. Harper, D. Hiemstra, T. Hofmann, E. Hovy, W. Kraaij, J. Lafferty, V. Lavrenko, D. Lewis, L. Liddy, R. Manmatha, A. McCallum, J. Ponte, J. Prager, D. Radev, P. Resnik, S. Robertson, R. Rosenfeld, S. Roukos, M. Sanderson, R. Schwartz, A. Singhal, A. Smeaton, H. Turtle, E. Voorhees, R. Weischedel, J. Xu, and C. Zhai. Challenges in Information Retrieval and Language Modeling: Report of a Workshop held at the Center for Intelligent Information Retrieval, University of Massachusetts Amherst, September 2002. *SIGIR Forum*, 37:31–47, April 2003. ISSN 0163-5840.

Omar Alonso and Stefano Mizzaro. Can we get rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In *SIGIR '09: Workshop on The Future of IR Evaluation*, 2009.

M. Amini and N. Usunier. A Contextual Query Expansion Approach by Term Clustering for Robust Text Summarization. In *Proceedings of the 7th Document Understanding Conference*, pages 48–55, Rochester, USA, 2007. DUC.

R. Angheluta, R. Mitra, X. Jing, and M. Moens. K.U.Leuven summarization system at DUC 2004. In *Proceedings of the 4th Document Understanding Conference*. DUC, 2004.

G. Armano, A. Giuliani, and E. Vargiu. Using snippets in text summarization: a comparative study and an application. In *Proceedings of the 3rd Italian Information Retrieval (IIR) Workshop*, volume 835 of *CEUR Workshop Proceedings*, pages 121–132. CEUR-WS.org, 2012.

M. Attia. Arabic Tokenization System. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*,

Semitic '07, pages 65–72, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

A. Azmi and S. Al-Thanyyan. A Text Summarizer for Arabic. *Computer Speech and Language*, 26(4):260–273, August 2012. ISSN 0885-2308.

S. Azzam, K. Humphreys, and R. Gaizauskas. Using Coreference Chains For Text Summarisation. In *ACL Workshop on Coreference and its Applications*. Association for Computational Linguistics, 1999.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval - the Concepts and Technology Behind Search, Second Edition*. Pearson Education Ltd., Harlow, England, 2011. ISBN 978-0-321-41691-9.

W. Banzhaf, F. Francone, R. Keller, and P. Nordin. *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 1-55860-510-X.

A. Barrera and R. Verma. Automated Extractive Single-document Summarization: Beating the Baselines with a New Approach. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC'11, pages 268–269, TaiChung, Taiwan, 2011. ACM. ISBN 978-1-4503-0113-8.

R. Barzilay, N. Elhadad, and K. McKeown. Sentence Ordering in Multidocument Summarization. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT'01, pages 1–7, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

A. Bawakid and M. Oussalah. A Semantic Summarization System: University of

Birmingham at TAC. In *The Proceedings of the Text Analysis Conference*. TAC, 2008.

P. Baxendale. Machine-made index for technical literature: an experiment. *IBM Journal of Research and Development*, 2(4):354–361, 1958. ISSN 0018-8646.

K. Beesley. Arabic Morphology Using only Finite-state Operations. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, Semitic '98, pages 50–57, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

Y. Benajiba, M. Diab, and P. Rosso. Arabic Named Entity Recognition: A Feature-Driven Study. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17 (5):926 –934, july 2009. ISSN 1558-7916.

E. Benmamoun. The Syntax of Arabic Tense. *Cahiers de Linguistique de L'INALCO*, 5:9–25, 2007. ISSN 1298–9851.

A. Berger and V. Mittal. Query-relevant Summarization using FAQs. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 294–301, Morristown, NJ, USA, 2000. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075218.1075256.

S. Blair-Goldensohn and K. McKeown. Integrating rhetorical-semantic relation models for query-focused summarization. In *Proceedings of 6th Document Understanding Conference*. DUC, 2006.

C. Blake, J. Kampov, A. Orphanides, D. West, and C. Lown. UNCCH at DUC 2007: Query expansion, Lexical Simplification and Sentence Selection Strategies for Multi-Document Summarization. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

B. Bollobas. *Modern Graph Theory*. Springer, corrected edition, July 1998. ISBN 0387984887.

A. Bossard and C. Rodrigues. Combining a Multi-Document Update Summarization System CBSEAS with a Genetic Algorithm. In *International Workshop on Combinations of Intelligend Methods and Applications*, CIMA 2010, Arras, France, 2010. Hyper Articles en Ligne.

M. Boudabous, M. Maaloul, and L. Belguith. Digital Learning for Summarizing Arabic Documents. In *Proceedings of the 7th International Conference on Advances in Natural Language Processing*, IceTAL'10, pages 79–84, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14769-0, 978-3-642-14769-2.

F. Boudin, F. Béchet, M. El-Béze, B. Favre, L. Gillard, and J. Torres-Moreno. The LIA Summarization System at DUC 2007. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

O. Boydell and B. Smyth. From social bookmarking to social summarization: An experiment in community-based summary generation. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, IUI '07, pages 42–51, New York, NY, USA, 2007. ACM. ISBN 1-59593-481-2.

R. Brandow, K. Mitze, and Lisa F. Rau. Automatic Condensation of Electronic Publications by Sentence Selection. *Inf. Process. Manage.*, 31(5):675–685, 1995. ISSN 0306-4573.

O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the Whole in Parts: Text Summarization For Web Browsing on Handheld Devices. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 652–662, New York, NY, USA, 2001. ACM. ISBN 1-58113-348-0.

J. Carbonell and J. Goldstein. 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, pages 335–336, New York, NY, USA, 1998. ACM.

P. Chen and R. Verma. A Query-Based Medical Information Summarization System Using Ontology Knowledge. In *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems CBMS'06*, pages 37–42, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2517-1.

C. Chowdary and P. Kumar. Update Summariser Using MMR Approach. In *The Proceedings of the Text Analysis Conference*. TAC, 2008.

J. Conroy and J. Schlesinger. CLASSY and TAC 2008 Metrics. In *In the Proceedings of the Text Analysis Conference*. TAC, 2008.

J. Conroy, D. Schlesinger, and G. Stewart. Classy Query-Based Multi-Document Summarization. In *Proceedings of the 5th Document Understanding Conferences*. DUC, 2005.

J. Conroy, J. Schlesinger, D. O'Leary, and J. Goldstein. Back to Basics: CLASSY 2006. In *Proceedings of the 6th Document Understanding Conferences*. DUC, 2006.

C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, September 1995. ISSN 0885-6125.

B. Croft, D. Metzler, and T. Strohman. *Search Engines - Information Retrieval in Practice*. Pearson Education, 2009. ISBN 978-0-13-136489-9.

H. Dang. Overview of DUC (2007). In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

H. Dang and K. Owczarzak. Overview of the TAC 2008 Update Summarization Task. In *Text Analysis Conference (TAC) 2008*, pages 10–23, Maryland, USA, 2008. TAC.

K. Darwish, H. Hassan, and O. Emam. Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Semitic '05, pages 25–30, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

E. D'Avanzo and B. Magnini. A Keyphrase-Based Approach To Summarization : The Lake System. In *Proceedings of the 5th Document Understanding Conferences*. DUC, 2005.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

M. Diab, K. Hacioglu, and D. Jurafsky. Automatic Processing of Modern Standard Arabic Text. In A. Soudi, A. van den Bosch, and G. Neumann, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, Text, Speech and Language Technology, pages 159–179. Springer Netherlands, 2007.

L. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945.

F. Diehl, M. Gales, M. Tomalin, and P. Woodland. Morphological Decomposition in Arabic ASR Systems. *Computer Speech and Language*, 26(4):229–243, August 2012. ISSN 0885-2308.

E. Diemert and G. Vandelle. Unsupervised Query Categorization Using Automatically-Built Concept Graphs. In *Proceedings of the 18th international conference on World*

Wide Web WWW'09, pages 461–470, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4.

R. Donaway, K. Drummey, and L. Mather. A Comparison of Rankings Produced by Summarization Evaluation Measures. In *NAACL-ANLP 2000 Workshop on Automatic Summarization*, pages 69–78, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

F. Douzidia and G. Lapalme. Lakhas, an Arabic Summarising System. In *Proceedings of the 4th Document Understanding Conferences* , pages 128–135. DUC, 2004.

S. Dumais, G. Furnas, T. Landauer, S. Deerwester, and R. Harshman. Using Latent Semantic Analysis to Improve Access to Textual Information. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems (CHI'88)*, pages 281–285. ACM, 1988.

D. Dunlavy, D. O'Leary, J. Conroy, and J. Schlesinger. QCS: A System for Querying, Clustering and Summarizing Documents. *Information Processing and Management*, 43:1588–1605, November 2007. ISSN 0306-4573.

H. Edmundson. New Methods in Automatic Extracting. *J. ACM*, 16(2):264–285, 1969. ISSN 0004-5411.

M. El-Haj. Experimenting with Automatic Summarization of Arabic Text. In *MSc. Thesis in Information Systems at King Abdullah II School of Information Technology*, Amman, Jordan, 2008. The University of Jordan.

M. El-Haj, U. Kruschwitz, and C. Fox. Experimenting with Automatic Text Summarization for Arabic. In Zygmunt Vetulani, editor, *4th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science*

and Linguistics, LTC'09, "Lecture Notes in Artificial Intelligence", pages 490–499, Poznan, Poland, 2009. Springer.

M. El-Haj, U. Kruschwitz, and C. Fox. Using Mechanical Turk to Create a Corpus of Arabic Summaries. In *Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages workshop held in conjunction with the 7th International Language Resources and Evaluation Conference (LREC 2010).*, pages 36–39, Valletta, Malta, 2010. LREC 2010.

M. El-Haj, U. Kruschwitz, and C. Fox. Multi-Document Arabic Text Summarisation. In *The 3rd Computer Science and Electronic Engineering Conference (CEEC'11)*, Colchester, UK, 2011a. IEEE Xplore.

M. El-Haj, U. Kruschwitz, and C. Fox. Exploring Clustering for Multi-Document Arabic Summarisation. In M. Salem, K. Shaalan, F. Oroumchian, A. Shakery, and H. Khelalfa, editors, *The 7th Asian Information Retrieval Societies (AIRS 2011)*, volume 7097 of *Lecture Notes in Computer Science*, pages 550–561. Springer Berlin / Heidelberg, 2011b. ISBN 978-3-642-25630-1.

M. El-Haj, U. Kruschwitz, and C. Fox. University of Essex at the TAC 2011 Multilingual Summarisation Pilot. In *Text Analysis Conference (TAC) 2011, MultiLing Summarisation Pilot*, Maryland, USA, 2011c. TAC.

B. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, West Susses, UK, 2005. ISBN 978-0-470-74991-3.

M. Fattah and F. Ren. Automatic Text Summarization. In *Proceedings of World Academy of Science*, volume 27, pages 192–195. World Academy of Science, 2008.

C. Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech,*

and Communication). The MIT Press, Illustrated Edition edition, May 1998. ISBN 026206197X.

K. Filippova, M. Mieskes, V. Nastase, S. Ponzetto, and M. Strube. Cascaded Filtering for Topic-Driven Multi-Document Summarization. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

S. Fisher and B. Roark. Query-Focused Summarization By Supervised Sentence Ranking and Skewed Word Distributions. In *Proceedings of the 6th Document Understanding Conferences* . DUC, 2006.

S. Fisher and B. Roark. Feature Expansion for Query-Focused Supervised Sentence Ranking. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

M. Fiszman, D. Demner-Fushman, H. Kilicoglu, and T. Rindflesch. Automatic Summarization of MEDLINE Citations for Evidence-based Medical Treatment: A Topic-oriented Evaluation. *Jouranl of Biomedical Informatics*, 42(5):801–813, 2009. ISSN 1532-0464.

J. Flores, L. Gillard, O. Ferret, and G. Chalendar. Bag-of-Senses Versus Bag-of-Words: Comparing Semantic and Lexical Approaches on Sentence Extraction. In *The Proceedings of the Text Analysis Conference Workshop*. TAC, 2008.

C. Fox. A Stop List for General Text. *SIGIR Forum*, 24:19–21, September 1989. ISSN 0163-5840.

F. Fukumoto, A. Sakai, and Y. Suzuki. Eliminating Redundancy by Spectral Relaxation for Multi-document Summarization. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-5, pages 98–

102, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-77-0.

P. Fung and G. Ngai. One Story, One Flow: Hidden Markov Story Models for Multilingual Multidocument Summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–16, July 2006.

A. Funk, D. Maynard, H. Saggion, and K. Bontcheva. Ontological Integration of Information Extracted from Multiple Sources. In *The Multi-source Multilingual Information Extraction and Summarization (MMIES) workshop at Recent Advances in Natural Language Processing (RANLP07)*, Borovets, Bulgaria, 2007. RANLP07.

D. Galanis and P. Malakasiotis. AUEB at TAC 2008. In *The Proceedings of the Text Analysis Conference*. TAC, 2008.

A. Ganapathi and S. Zhang. Web Analytics and the Art of Data Summarization. In *Managing Large-scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques*, SLAML '11, pages 6:1–6:9, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0978-3.

A. Gancarski, A. Doucet, and P. Henriques. Attribute Grammar-based Interactive System to Retrieve Information from XML Documents. *IEE Proceedings - Software*, 153(2):51–60, 2006.

G. Giannakopoulos and V. Karkaletsis. AutoSummENG and MeMoG in Evaluating Guided Summaries. In *The Proceedings of the Text Analysis Conference*, MD, USA, 2011. TAC.

G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos. Summarization System Evaluation Revisited: N–Gram Graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):1–39, 2008.

G. Giannakopoulos, M. El-Haj, B. Favre, M. Litvak, J. Steinberger, and V. Varma. TAC 2011 MultiLing Pilot Overview. In *Text Analysis Conference (TAC) 2011, MultiLing Summarisation Pilot*, Maryland, USA, 2011. TAC.

J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitzt. Multi-Document Summarization by Sentence Extraction. In *NAACL-ANLP Workshop on Automatic Summarization*, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

F. Gotti, G. Lapalme, L. Nerima, and E. Wehrli. GOFAISUM: A Sympolic Summarizer for DUC. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

A. Grewal, T. Allison, S. Dimitrov, and D. Radev. Multi-Document Summarization Using Off the Shelf Compression Software, 2003.

D. Grossman and O. Frieder. *Information Retrieval: Algorithms and Heuristics*. The Kluwer International Series of Information Retrieval. Springer, second edition, 2004.

E. Günes. Using Biased Random Walks For Focused Summarization. In *Proceedings of the 6th Document Understanding Conferences*. DUC, 2006.

Y. Guo and G. Stylios. A New Multi-Document Summarization System. In *Proceedings of the 4th Document Understanding Conferences*. DUC, 2004.

N. Habash and R. Roth. Using Deep Morphology to Improve Automatic Error Detection in Arabic Handwriting Recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 875–884, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

B. Hachey, G. Murray, and D. Reitter. Query-oriented multi-document summarization

with a very large latent semantic space. the embra system. In *Proceedings of the 5th Document Understanding Conferences*. DUC, 2005.

B. Haddad and M. Yaseen. A Compositional Approach Towards Semantic Representation and Construction of ARABIC. In *Proceedings of the 5th International Conference on Logical Aspects of Computational Linguistics*, LACL'05, pages 147–161, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-25783-7, 978-3-540-25783-7.

L. Hadrich, L. Baccour, and G. Mourad. Segmentation of Arabic Texts Based on Contextual Analysis of Punctuation marks and Certain Particle. In *Proceedings of the 12th conference on Computational Natural Language*, pages 451–456, Dourdan-France, 2005. TALN'2005.

M. Hajime and O. Manabu. A Comparison of Summarization Methods based on Task-based Evaluation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece, 2000. LREC.

K. Hammouda and M. Kamel. Efficient Phrase-Based Document Indexing for Web Document Clustering. *IEEE Transaction on Knowledge and Data Engineering*, 16: 1279–1296, October 2004. ISSN 1041-4347.

K. Han, Y. Song, and H. Rim. KU Text Summarization System for DUC 2003. In *Proceedings of the 4th Document Understanding Conferences*. DUC, 2004.

T. He, J. Chen, and Z. Gui F. Li. CCNU at TAC 2008: Proceeding on Using Semantic Method for Automated Summarization Yield. In *Proceedings of the Text Analysis Conference*. TAC, 2008.

I. Hendrickx, W. Daelemans, E. Marsi, and E. Krahmer. Reducing Redundancy in Multi-document Summarization using Lexical Semantic Similarity. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, UCNLG+Sum

'09, pages 63–66, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-51-0.

A. Hickl, K. Roberts, and F. Lacatusui. LCC's GISTexter at DUC 2007: Machine reading for update summarization. In *Proceedings of the 7th Document Understanding Conferences* . DUC, 2007.

T. Hirao, J. Suzuki, H. Isozaki, and E. Maeda. Ntt's Multiple Document Summarization System. In *Proceedings of the 4th Document Understanding Conferences*. DUC, 2004.

T. Hirao, M. Okumura, N. Yasuda, and H. Isozaki. Supervised Automatic Evaluation for Summarization with Voted Regression Model. *Information Processing and Management*, 43(6):1521–1535, November 2007. ISSN 0306-4573.

I. Hmeidi, R. Al-Shalabi, A. Al-Taani, H. Najadat, and S. Al-Hazaimeh. A Novel Approach to the Extraction of Roots from Arabic Words using Bigrams. *Journal of the American Society for Information Science and Technology*, 61(3):583–591, 2010. ISSN 1532-2890.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1.

E. Hovy and C. Lin. Automated Text Summarization and the SUMMARIST System. In *Proceedings of a Workshop on Held at Baltimore, Maryland*, pages 197–214, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

H. Huang, Y. Kuo, and H. Yang. Fuzzy–Rough Set Aided Sentence Extraction Summarization. In *Proceedings of the First International Conference on Innovative Comput-*

ing, Information and Control - Volume 1, ICICIC'06, pages 450–453, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2616-0.

L. Jaqua, M. Jaoua, H. Belguith, and H. Ben. Summarization at LARIS laboratory. In *Proceedings of the 4th Document Understanding Conferences*. DUC, 2004.

Z. Jin, C. Xueqi, X. Hongbo, W. Xiaolei, and Z. Yiling. ICTCAS's ICTGrasper at TAC 2008: Summarizing Dynamic Information with Signature Terms Based Content Filtering. In *Proceedings of the Text Analysis Conference*. TAC, 2008.

M. Kan. *Automatic Text Summarization as Applied to Information Retrieval: using Indicative and Informative Summaries*. PhD thesis, Columbia University, New York, NY, USA, 2003. AAI3071379.

R. Katragadda, P. Pingali, and V. Varma. Sentence Position Revisited: A Robust Light–Weight Update Summarization 'Baseline' Algorithm. In *Proceedings of the Third International Workshop on Cross Lingual Information Access CLIAWS3'09*, pages 46–52, Morristown, NJ, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-33-6.

S. Katz. Distribution of content words and phrases in text and language modelling. *Nat. Lang. Eng.*, 2(1):15–59, March 1996. ISSN 1351-3249.

G. Kazai, J. Kamps, M. Koolen, and N. Milic-Frayling. Crowdsourcing for Book Search Evaluation: Impact of Hit Design on Comparative System Ranking. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 205–214, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4.

S. Khoja and S. Garside. Stemming Arabic Text, 1999.

L. Khreisat. Arabic Text Classification using N-gram Frequency Statistics: A Comparative Study. In *Proceedings of the 2006 International Conference on Data Mining*, pages 78–82, 2006.

A. Kittur, B. Smus, S. Khamkar, and R. Kraut. CrowdForge: CrowdSourcing Complex Work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 43–52, New York, NY, USA, 2011. ACM.

K. Knight and D. Marcu. Statistics-Based Summarization – Step One: Sentence Compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703–710, Menlo Park, CA, 2000. AAAI Press. ISBN 0-262-51112-6.

M. Kolla, O. Vechtomova, and C. Clarke. Comparison of Models Based on Summaries or Documents Towards Extraction of Update summaries. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

C. Kruengkrai and C. Jaruskulchai. Generic Text Summarization Using Local and Global Properties of Sentences. In *Web Intelligence*, pages 201–206. IEEE Computer Society, 2003.

J. Kupiec, J. Pedersen, and F. Chen. A Trainable Document Summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 68–73, Seattle, Washington, United States, 1995. ACM Press.

F. Lacatusu, Y. Shi, J. Bensley, B. Rink, P. Wang, and L. Taylor. Lcc's Gistexter: Multi-Strategy Multi-Document Summarization. In *Proceedings of the 6th Document Understanding Conferences*. DUC, 2006.

L. Larkey, L. Ballesteros, and M. Connell. Improving Stemming for Arabic Information

Retrieval: Light Stemming and Co-occurrence Analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 275–282, New York, NY, USA, 2002. ACM. ISBN 1-58113-561-0.

D. Leite and L. Rino. Combining Multiple Features for Automatic Text Summarization through Machine Learning. In António Teixeira, Vera Lúcia Strube de Lima, Luís Caldas de Oliveira, and Paulo Quaresma, editors, *Proceedings of the 8th international conference on Computational Processing of the Portuguese Language (PROPOR'08)*, volume 5190 of *Lecture Notes in Computer Science*, pages 122–132. Springer, 2008.

D. Lemire. Faster Retrieval with a Two-Pass Dynamic-Time-Warping Lower Bound. *Pattern Recognition*, 42(9):2169–2180, September 2009. ISSN 0031-3203.

J. Li, L. Kit, and J. Webster. A Query-Focused Multi-Document Summarizer Based on Lexical Chains. In *Proceedings of the 6th Document Understanding Conferences*. DUC, 2006.

J. Li, W. Wang, and C. Wang. TAC 2008 Update Summarization Task of ICL. In *Proceedings of the Text Analysis Conference*. TAC, 2008.

S. Li, Y. Ouyang, W. Wang, and B. Sun. Multi-Document Summarization Using Support Vector Regression. In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

C. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26. WAS 2004), 2004.

C. Lin and E. Hovy. From Single to Multi-document Summarization: A Prototype

System and its Evaluation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 07–12, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

F. Liu and Y. Liu. Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short'08, pages 201–204, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

S. Liu and J. Lindroos. Towards Fast Digestion of IMF Staff Reports with Automated Text Summarization Systems. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 978–982. IEEE Computer Society, 2006.

X. Liu and B. Croft. Cluster-based Retrieval using Language Models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 186–193, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4.

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, pages 129–136, 1982.

J. Lovins. Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

H. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

N. Madnani, D. Zajic, B. Dorr, N. Ayan, and J. Lin. Multiple Alternative Sentence

Compressions for Automatic Text Summarization. In *Proceedings of the 7th Document Understanding Conference at NLT/NAACL*, page 26. DUC, 2007.

B. Maegaard, M. Atiyya, K. Choukri, S. Krauwer, C. Mokbel, and M. Yaseen. Medar: Collaboration between European and Mediterranean Arabic Partners to Support the Development of Language Technology for Arabic. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008. LREC.

D. Mallett, J. Elding, and M. Nascimento. Information-Content Based Sentence Extraction for Text Summarization. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume*, volume 2, page 214, Washington, DC, USA, 2004. IEEE Computer Society.

W. Mann and S. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281, 1988.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

K. McKeown and D. Radev. Generating Summaries of Multiple News Articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'95, pages 74–82, New York, NY, USA, 1995. ACM. ISBN 0-89791-714-6.

R. Mihalcea. Language Independent Extractive Summarization. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 49–52, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

R. Mihalcea and P. Tarau. Multi-Document Summarization with Iterative Graph-based Algorithms. In *The First International Conference on Intelligent Analysis Methods and Tools (IA 2005)*, McLean, VA, 2005. UNT Digital Library.

H. Morita, T. Sakai, and M. Okumura. Query Snowball: A Co-occurrence-based Approach to Multi-document Summarization for Question Answering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT'11, pages 223–229, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-88-6.

J. Myers and A. Well. *Research Design and Statistical Analysis*. Routledge, New Jersey, 2 edition, 2003. ISBN 978-0805840377.

V. Nastase, K. Filippova, and S. Ponzetto. Generating Update Summaries with Spreading Activation. In *Proceedings of the Text Analysis Conference*. TAC, 2008.

E. Newman, W. Doran, N. Stokes, J. Carthy, and J. Dunnion. Comparing Redundancy Removal Techniques for Multi-document Summarisation. *Proceedings of the Second Starting AI Researchers' Symposium (Stairs'04)*, 2004.

R. Nielsen. Question generation: Proposed challenge tasks and their evaluation. In *Proceeding of the Question Generation Shared Task and Evaluation Challenge Workshop*, 2008.

T. Nomoto and Y. Matsumoto. A New Approach to Unsupervised Text Summarization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 26–34, New York, NY, USA, 2001. ACM.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: A Method for Automatic

Evaluation of Machine Translation. In *Proceeding of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*. Association for Computational Linguistics, 2002.

M. Porter. *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5.

S. Prochazka. *Arabic. Encyclopedia of Language and Linguistics*, volume 1. Elsevier, 2nd edition, 2006.

Y. Qiu and H. Frei. Concept Based Query Expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 160–169, New York, NY, USA, 1993. ACM.

D. Radev, H. Jing, and M. Budzikowska. Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and User Studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization – Volume 4*, NAACL-ANLP-AutoSum '00, pages 21–30, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

D. Radev, H. Jing, M. Sty, and D. Tam. Centroid-based Summarization of Multiple Documents. *Information Processing and Management*, 40:919–938, 2004. doi: 10.1016/j.ipm.2003.10.006.

F. Rasheed, Y. Lee, and S. Lee. Applying Context Summarization Techniques in Pervasive Computing Systems. In *Proceedings of the The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)*, SEUS-WCCIA '06, page 6. IEEE Computer Society, 2006.

A. Roberts, L. Al-Sulaiti, and ES Atwell. aConCorde: Towards an Open-source, Extendable Concordancer for Arabic. *Corpora*, 1(1):39–60, 2006. doi: 10.3366/cor.2006.1.1.39.

H. Salhi. Small Parallel Corpora in an English-Arabic Translation Classroom: No Need to Reinvent the Wheel in the Era of Globalization. In *Globalisation and Aspects of Translation*, pages 53–67, UK, 2010. Newcastle: Cambridge Scholars Publishing.

G. Salton. *Automatic Text processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0-201-12227-8.

G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.

G. Salton, A. Wong, and S. Yang. A Vector Space Model for Automatic Indexing. *Proceedings of the Communications of the ACM*, 18(11):613–620, 1975.

M. Saravanan, S. Raman, and B. Ravindran. A Probabilistic Approach to Multi-Document Summarization for Generating a Tiled Summary. In *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications*, ICCIMA '05, pages 167–172, Washington, DC, USA, 2005. IEEE Computer Society.

K. Sarkar. Centroid-based summarization of multiple documents. *TECHNIA – International Journal of Computing Science and Communication Technologies*, 2, 2009.

M. Sawalha and ES. Atwell. Constructing and Using Broad-coverage Lexical Resource for Enhancing Morphological Analysis of Arabic. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Ros-

ner, and Daniel Tapias, editors, *The 7th Language Resources and Evaluation Conference LREC*, pages 282–287, Valletta, Malta, 2010a. LREC 2010. ISBN 2-9517408-6-7.

M. Sawalha and ES. Atwell. Fine-Grain Morphological Analyzer and Part-of-Speech Tagger for Arabic Text. In *The 7th Language Resources and Evaluation Conference LREC*, pages 1258–1265, Valletta, Malta, 2010b. LREC 2010.

B. Schiffman. Summarization for Q&A at Columbia University for DUC 2007. In *Proceedings of the 7th Document Understanding Conferences*. DUC, 2007.

F. Schilder, R. Kondadadi, J. Leidner, and J. Conrad. Thomson Reuters at TAC 2008: Aggressive Filtering with FastSum for Update and Opinion Summarization. In *Proceedings of the Text Analysis Conference*. TAC, 2008.

J. Schlesinger, D. O'Leary, and J. Conroy. Arabic/English Multi-Document Summarization With CLASSY: The Past and the Future. In *Proceedings of the 9th International Conference on Computational linguistics and Intelligent Text Processing*, CICLing'08, pages 568–581, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3-540-78134-X, 978-3-540-78134-9.

Y. Seki, K. Eguchi, N. Kando, and M. Aono. Multi-document Summarization with Subjectivity Analysis at DUC 2005. In *Proceedings of the 5th Document Understanding Conference*. DUC, 2005.

Y. Seki, K. Eguchi, N. Kando, and M. Aono. Opinion-Focused Summarization and its Analysis at DUC 2006. In *Proceedings of the 6th Document Understanding Conference*. DUC, 2006.

A. Singhal, C. Buckley, and M. Mitra. Pivoted Document Length Normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research*

and Development in Information Retrieval, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8.

J. Sjöbergh. Older Versions of the ROUGEeval Summarization Evaluation System were Easier to Fool. *Information Processing and Management*, 43(6):1500–1505, November 2007. ISSN 0306-4573.

D. Sleator and D. Temperley. Parsing English with a Link Grammar. In *In Third International Workshop on Parsing Technologies*, 1991.

O. Smrž. ElixirFM: Implementation of Functional Arabic Morphology. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, Semitic '07, pages 1–8, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

N. Stokes, J. Rong, and L. Cavedon. NICTA's Update and Question-based Summarisation Systems at DUC 2007. In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

Q. Su, D. Pavlov, J. Chow, and W. Baker. Internet-scale Collection of Human-reviewed Data. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 231–240, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7.

R. Sureka and H. Kong. Automated Trainable Summarizer For Financial Documents. In *EDOC Workshops*, page 55. IEEE Computer Society, 2006.

K. Svore. Enhancing Single-document Summarization by Combining RankNet and Third-party Sources. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 448–457. Association for Computational Linguistics, 2007.

K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. The PYTHY Summarization System: Microsoft Research at DUC 2007. In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

M. Turchi, J. Steinberger, M. Kabadjov, and R. Steinberger. Using Parallel Corpora for Multilingual (Multi-Document) Summarisation Evaluation. In *Proceedings of the 2010 International Conference on Multilingual and Multimodal Information Access Evaluation: Cross-language Evaluation Forum*, CLEF'10, pages 52–63, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15997-4, 978-3-642-15997-8.

L. Vanderwende, H. Suzuki, and C. Brockett. Task-Focused Summarization with Sentence Simplification and Lexical Expansion. In *Proceedings of the 6th Document Understanding Conference*. DUC, 2006.

R. Verma, P. Chen, and W. Lu. A Semantic Free-Text Summarization System Using Ontology Knowledge. In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

S. Wan and C. Paris. Experimenting with Clause Segmentation for Text Summarisation. In *Proceedings of the Text Analysis Conference*. TAC, 2008.

X. Wan and J. Yang. Multi-Document Summarization using Cluster-based Link Analysis. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'08, pages 299–306, Singapore, Singapore, 2008. ACM. ISBN 978-1-60558-164-4.

D. Wang and T. Li. Weighted Consensus Multi-document Summarization. *Information Processing and Management*, 2011. ISSN 0306-4573.

M. White and C. Cardie. Selecting Sentences for Multi-Document Summaries Using Randomized Local Search, July 2002.

J. Wightwick and M. Gaafar. *Arabic Verbs and Essentials of Grammar: a Practical Guide to the Mastery of Arabic*. Verbs and Essentials of Series. Passport Books, 1998. ISBN 9780844246055.

R. Witte, R. Krestel, and S. Bergle. Generating update summaries for DUC 2007. In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

C. Wolf, S. Alpert, J. Vergo, L. Kozakov, and Y. Doganata. Summarizing Technical Support Documents for Search: Expert and user Studies. *IBM Systems Journal*, 43 (3):564–586, 2004. ISSN 0018-8670.

Y. Wu, K. Chang, Y. Lee, and J. Yang. Light-Weight Multi-Document Summarization Based On Two-Pass Reranking. In *Proceedings of the 6th Document Understanding Conference*. DUC, 2006.

Z. Xiang and D. Fesenmaier. Assessing the initial step in the persuasion process: Meta tags on destination marketing websites. In Andrew J. Frew, editor, *Information and Communication Technologies in Tourism 2005*, pages 215–226. Springer Vienna, 2005. ISBN 978-3-211-27283-1.

H. Yan, W. Grosky, and F. Fotouhi. Augmenting the Power of LSI in Text Retrieval: Singular Value Rescaling. *Data Knowl. Eng.*, 65(1):108–125, 2008. ISSN 0169-023X.

M. Yaseen and N. Theophilopoulos. NAPLUS: Natural Arabic Processing for Language Understanding Systems, 2001.

J. Yeh, H. Ke, and W. Yang. ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications*, 35(3):1451–1462, 2008. ISSN 0957-4174.

J. Ying, S. Yen, Y. Lee nad Y. Wu, and J. Yang. Language Model Passage Retrieval for Question-Oriented Multi Document Summarization. In *Proceedings of the 7th Document Understanding Conference*. DUC, 2007.

M. Zajic and J. Lin. Sentence Compression as a Component of a Multi-Document Summarization System. In *Proceedings of the 6th Document Understanding Conference*. DUC, 2006.

H. Zha. Generic Summarization and Keyphrase Extraction using Mutual Reinforcement Principle and Sentence Clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'02, pages 113–120, New York, NY, USA, 2002. ACM. ISBN 1-58113-561-0.

L. Zhao, L. Wu, and X. Huang. Using Query Expansion in Graph-Based Approach for Query-Focused Multi-Document Summarization. *Inf. Process. Manage.*, 45(1): 35–41, 2009. ISSN 0306-4573.

Q. Zhou, L. Sun, and J. Nie. Is–Sum: A Multi-Document Summariser Based On Document Index Graphic and Lexical Chains. In *Proceedings of the 5th Document Understanding Conference*. DUC, 2005.

Appendices

Appendix A

EASC Corpus Guidelines Appendix

Creating EASC Corpus Guidelines

The Mechanical Turk workers were given the following guidelines for completing the task of creating single-document summaries corpus (Chapter 4).

1. Read the Arabic sentences (Document).
2. In the first text box below type the number of sentences that you think are focusing on the main idea of the document.
3. The number of the selected sentences should not exceed 50% of the articles‘ sentences, for example if there are 5 sentences you can only chose up to 2 sentences.
4. Just add the numbers of the sentences, please do not write text in the first text box. For example (1,2,4).
5. In the second text box write down one to three Keyword(s) that represent the main idea of the documents, or keywords that can be used as title for the article, do not exceed 3 keywords.
6. If you have any comments please add them to the third text box below.

7. Failing to follow the guides correctly could lead to task rejection

8. NOTE: The article chosen for this task was selected randomly from the Internet. The purpose of this task is purely educational and does not reflect, support or contradict with any opinion or point of view.

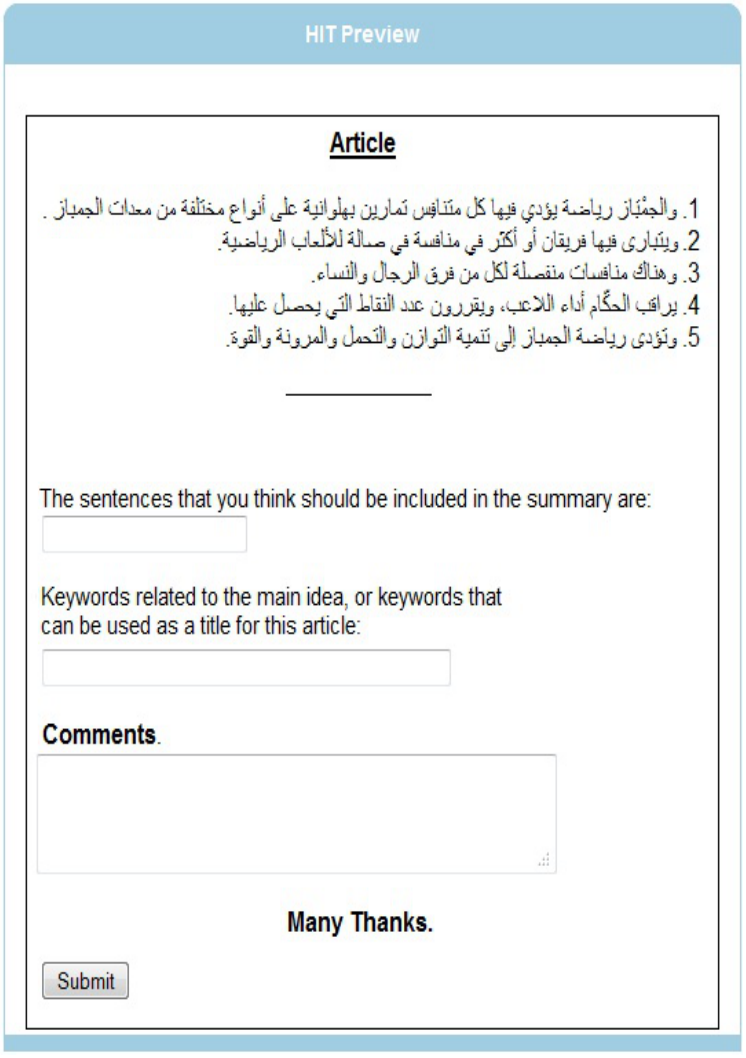

Figure A.1: EASC: MTurk Hit Example

Figure A.1, shows an example for one of the hits provided to the workers on Mechanical Turk website. The reason for the selection of an answer field (instead of a checkbox or radio button) is that we aimed to reduce the noise and track spammers.

We think that a design with e.g. radio buttons is not able to distinguish between MTurks as spammers and MTurks who produce noise (wrong selection but not produced by random procedure as it is the case by spammers), for example if the worker wrote “two” instead of “2”, we still consider this as a valid answer. Using radio buttons, when selection is made it is not clear whether the MTurk worker has selected it because he thought it is a correct answer or just by random. However, if we force an MTurk to write down the answer then this gives us the possibility to distinguish between spammers and noise. A spammer would give answers which are composed by random characters and/or numbers whereas a noise could be close to the right answer.

Appendix B

TAC–2011 Dataset Guidelines

Appendix

Creating TAC–2011 Dataset Guidelines

The following task guidelines were required by the participants to create a manual corpus for TAC–2011 MultiLing Pilot:

1. **Translation:** Given the source language text A, the translator is requested to translate each sentence in A, into the target language. Each target sentence should keep the meaning from the source language. The resulting text would be a UTF8 encoded plain text file, named $A.[lang]$, where $[lang]$ should be replaced by the target language. For each text the following check list should be followed:

    - The translator notes down the starting time for the reading step.
    - The translator reads the source text at least once to get an understanding.
    - The translator notes down the starting time for the translation step.
    - Perform the translation.
    - The translator notes down the finishing time for the translation step.

2. **Translation Validation:** After the completion of each translation, another translator "validator" should verify the correctness of the output. If errors are found, then the validator is to perform any corrections and finalise the translation. For each text the following check list should be followed:

   - The translator notes down the starting time for the verification step.
   - Read the translation and verify the text. Perform any corrections needed.
   - The translator notes down the finishing time for the verification step.

3. **Summarisation:** The summariser will read the whole set of texts at least once. Then, the summariser should compose a summary, with a minimum size of 240 and a maximum size of 250 words. The summary should be in the same language as the texts in the set. The aim is to create a summary that covers all the major points of the document set (what is major is left to summariser discretion). The summary should be written using fluent, easily readable language. No formatting or other markup should be included in the text. The output summary should be a self–sufficient, clearly written text, providing no other information than what is included in the source documents. For each document set the following check list should be followed:

   - The summariser notes down the starting time for the reading step.
   - Read the whole set of texts in the document set at least once, to have an overall understanding of the event(s) described.
   - The summariser notes down the starting time for the summarisation step.
   - The summariser writes the summary, reviewing the source texts, if required.
   - The summariser notes down the end time for the summarisation step.

4. **Evaluation:** Each summary will be graded by 3 evaluators. If the summarisers are used as evaluators, no self-evaluation should be allowed. Evaluators read

each translated document set at least once. Then they read the summary they are to evaluate, and they grade it. Each summary is to be assigned an integer grade from 1 to 5, related to the overall responsiveness of the summary. We consider a text to be worth a 5, if it appears to cover all the important aspects of the corresponding document set using fluent, readable language. A text should be assigned a 1, if it is either unreadable, nonsensical, or contains only trivial information from the document set. We consider the content and the quality of the language to be equally important in the grading.