Empirical Evaluation Methodology for Target Dependent Sentiment Analysis

Andrew Phillip Moore B.Sc. (Hons)
School of Computing and Communications
Lancaster University

A thesis submitted for the degree of
Doctor of Philosophy

July, 2020

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is: 59825

Andrew Phillip Moore

Empirical Evaluation Methodology for Target Dependent Sentiment Analysis

Andrew Phillip Moore B.Sc. (Hons).
School of Computing and Communications, Lancaster University
A thesis submitted for the degree of *Doctor of Philosophy*. July, 2020.

Abstract

The area of sentiment analysis has been around for at least 20 years in one form or another. In which time, it has had many and varied applications ranging from predicting film successes to social media analytics, and it has gained widespread use via selling it as a tool through application programming interfaces. The focus of this thesis is not on the application side but rather on novel evaluation methodology for the most fine grained form of sentiment analysis, target dependent sentiment analysis (TDSA). TDSA has seen a recent upsurge but to date most research only evaluates on very similar datasets which limits the conclusions that can be drawn from it. Further, most research only marginally improves results, chasing the State Of The Art (SOTA), but these prior works cannot empirically show where their improvements come from beyond overall metrics and small qualitative examples. By performing an extensive literature review on the different granularities of sentiment analysis, coarse (document level) to fine grained, a new and extended definition of fine grained sentiment analysis, the hextuple, is created which removes ambiguities that can arise from the context. In addition, examples from the literature will be provided where studies are not able to be replicated nor reproduced.

This thesis includes the largest empirical analysis on six English datasets across multiple existing neural and non-neural methods, allowing for the methods to be tested for generalisability. In performing these experiments factors such as dataset size and sentiment class distribution determine whether neural or non-neural approaches are best, further finding that no method is generalisable. By formalising, analysing, and testing prior TDSA error splits, newly created error splits, and a new TDSA specific metric, a new empirical evaluation methodology has been created for TDSA. This evaluation methodology is then applied to multiple case studies to empirically justify improvements, such as position encoding, and show how contextualised word representation improves TDSA methods. From the first reproduction study in TDSA, it is believed that random seeds significantly affecting the neural method is the reason behind the difficulty in reproducing or replicating the original study results. Thus highlighting empirically for the first in TDSA the need for reporting multiple run results for neural methods, to allow for better reporting and improved evaluation. This thesis is fully reproducible through the codebases and Jupyter notebooks referenced, making it an executable thesis.

Publications

Only one publication, shown below, has been created directly from the thesis, from which large portions of this published work is used within chapter 3:

Andrew Moore and Paul Rayson (Aug. 2018). "Bringing replication and reproduction together with generalisability in NLP: Three reproduction studies for Target Dependent Sentiment Analysis". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1132–1144. URL: https://www.aclweb.org/anthology/C18-1097

The following publication have been generated while developing this thesis, and to an extent has guided the thesis into what it has become:

Henry Moss, **Andrew Moore**, David Leslie, and Paul Rayson (July 2019). "FIESTA: Fast IdEntification of State-of-The-Art models using adaptive bandit algorithms". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2920–2930. URL: https://www.aclweb.org/anthology/P19-1281

Andrew Moore and Paul Rayson (Aug. 2017). "Lancaster A at SemEval-2017 Task 5: Evaluation metrics matter: predicting sentiment from financial news headlines". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 581–585. URL: https://www.aclweb.org/anthology/S17-2095

Mahmoud El-Haj, Paul Rayson, Steve Young, **Andrew Moore**, Martin Walker, Thomas Schleicher, and Vasiliki Athanasakou (May 2016). "Learning Tone and Attribution for Financial Text Mining". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 1820–1825. URL: https://www.aclweb.org/anthology/L16-1287

Andrew Moore, Paul Edward Rayson, and Steven Eric Young (2016). "Domain adaptation using stock market prices to refine sentiment dictionaries". In: *Proceedings of the Emotion and Sentiment Analysis Workshop LREC 2016, Portorož, Slovenia*, pp. 63–66. URL: http://gsi.dit.upm.es/esa2016/Proceedings-ESA2016.pdf

Acknowledgements

This PhD would never have been completed without the support, encouragement, and guidance that my supervisor, Paul Rayson, has given and still does, and in which I will always be thankful for.

I would like to thank Gerald Kotonya for the pre-PhD chat, which motivated me throughout the PhD. Similarly I am thankful for the support, advice, and time that Steven Young has given throughout the PhD. Thank you also to all of the past and current members of the C28 office and the NLP group. Lancaster University has been a great place to study for the last eight years, which in most has been due to the friends and work colleagues I have got to know, and the spectacular scenery that is Lancashire.

This PhD would not have been possible without the funding from Lancaster University by an EPSRC Doctoral Training Grant.

Outside of the PhD I am thankful to Jake Brown for being such a great friend throughout the PhD.

Finally none of this would have been possible without the continual support and encouragement of my family June, Jonathan, and Christopher Moore this achievement is as much mine as it is theirs. A special thank you to Bella, my families German Shorthaired Pointer, who never fails to bring a smile to my face at least once a day.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Sentiment analysis is a large sub field within the larger research area of Natural Language Processing (NLP), it has been defined by Liu (2015) as the analysis of people's opinions towards entities in written text. This definition to some degree is slightly out of date as the field has grown to encompass more information than just written text, such as images, voice, and video[1] (Poria et al., 2016). However as this thesis only uses textual information, this definition is representative of sentiment analysis in the research presented here. Following this definition, people's opinions are represented by sentiment values such as positive, negative, and neutral. Further, the entities within the definition can be an organisation, person, object, or more broadly a concept or topic such as politics within Britain.

The applications and thus the motivation for sentiment analysis are many and varied, due to a large degree the massive quantities of unstructured textual data that the web contains. Application examples include understanding what audiences liked and disliked within films and trailers (Pereg et al., 2019), linking the sentiment of the text about a film over time with a film's success (Vecchio et al., 2018), and providing additional information for analysis on political tweets (Wang et al., 2017b). Further, many businesses sell sentiment analysis via an application programming interface (API) including Google[2], Microsoft[3], and Aylien[4]. This thesis is less focussed in the end application of sentiment analysis, but rather whether the empirical evaluation of the methods created for the most fine grained version, target dependent sentiment analysis (TDSA), can be improved.

The research area of TDSA has recently seen a surge[5] of new research most likely due to commercial applications. This is fantastic for any area of research, however current research, in the majority of cases, is only publishing positive results[6] on very similar

[1] The use of more than one source of information e.g. voice and text, is commonly known as multimodal.

[2] `https://cloud.google.com/natural-language`

[3] `https://azure.microsoft.com/en-gb/services/cognitive-services/text-analytics/`

[4] `https://aylien.com/text-analysis-platform/`

[5] The chart from Papers With Code, shown here `https://paperswithcode.com/sota/aspect-based-sentiment-analysis-on-semeval`, plots a paper's method and its metric score on a set of popular TDSA datasets over time. Note that the chart states it is for the task Aspect-Based Sentiment Analysis, but in this thesis the task is named TDSA, the difference between the two will be explained in chapter 2.

[6] This is in reference to negative publication bias, of which there is a NLP workshop at EMNLP 2020

datasets without any empirical or quantitative reasons for these positive results. Thus TDSA methods cannot be shown empirically to be generalisable, where in this thesis generalisable means that a method performs well[7] across many vastly different datasets. Further, evaluation within this generalisable setup is approached by training and evaluating a method on each dataset independently. By testing methods for generalisability, rather than on a few similar datasets, it is possible to find out whether one method performs well across the board in all circumstances, or the more likely case of which methods perform best for different dataset properties. Thus testing for generalisability would allow researchers to get a better understanding of their method, where the method can be improved, and based on the dataset properties if it is the best method to use.

Even though generalisability can show some reasons why a method may work well on one dataset compared to another, e.g. a method performs well on review data and not social media, this form of analysis cannot explain why one method is better than another within a dataset. Thus a more fine grained analysis that can test different phenomena within a dataset is required, especially as the difference in results between methods on some dataset are marginal. Some approaches for fine grained analysis already exist that can test different phenomena within TDSA through error splits[8] of a dataset. Within the existing literature, several different error splits have been suggested that probe different phenomena within TDSA (Nguyen et al., 2015; Wang et al., 2017a; He et al., 2018b; Yang et al., 2018). However, so far no work has performed a detailed analysis of these error splits, nor has any work examined how these error splits can be improved upon. By better understanding what these prior works' error splits probe, improvements could be made, and more rigorous evaluation methodology can be created for TDSA.

If the results from a paper are difficult or impossible to reproduce or replicate[9] to a large extent that research is pointless, and this has been expressed through the fictional tale of the Zigglebottom tagger (Pedersen, 2008). From a review of the literature it has been found that one particular Neural Network (NN) based TDSA method has been difficult to reproduce or replicate[10]. Within neural sequence labelling, Reimers et al. (2017) found that NN methods can produce significantly different results between multiple runs due to random initialisations. Thus as these prior NN works within TDSA have only reported single runs it is plausible that the reason for the lack of replication or reproducibility could be due to not reporting multiple runs. Therefore a review within reporting standards for NN TDSA approaches is required for both reproducibility and fair evaluation.

The main goal of this thesis is to improve evaluation and reporting within TDSA so that researchers can better understand and reproduce the methods they create. This thesis also goes one step further whereby instead of just theoretically and empirically showing how evaluation and reporting can be improved, the whole thesis is 'executable'

that focuses on negative results `https://insights-workshop.github.io/`.

[7] In this thesis a method performs 'well' when it performs either significantly better than all other methods or it is not significantly different to the other top performing methods.

[8] We define an error split as a method of splitting an existing dataset into subsets of data, whereby each subset satisfies a different condition of the split. For instance the error split suggested by Wang et al. (2017a) is conditioned on the number of unique sentiments in a text, in this case for datasets with two unique sentiment values there would be only two subsets of data, the first subset for texts with a single unique sentiment, and the second for texts with two unique sentiments.

[9] The definition of the differences between reproduce and replicate will be explained in chapter 3.

[10] See chapter 3 section 3.5.4.

as in all code to produce the findings within the thesis is reproducible through Jupyter notebooks[11] and related codebases. All codebases and notebooks that are relevant to different sections to produce the results within those sections are stated within footnotes as URLs attached to the relevant section headers. Lastly all empirical results within this thesis are on datasets within the English language[12].

1.2 Research Questions

Building on the motivation provided above, these are the research questions that the thesis will answer:

Research Question 1 *What lessons can be learned from reproducing a method within TDSA?*

Research Question 2 *How generalisable are existing methods within TDSA?*

Research Question 3 *What is an appropriate empirical evaluation methodology for TDSA?*

1.3 Contributions and Findings

- **The creation of a new definition, the hextuple, for fine grained sentiment analysis that directly extends the current definition by Liu (2015).**
  The extended fine grained sentiment analysis definition in comparison to the existing by Liu (2015) removes any ambiguity that arises from the context[13]. This ambiguity is motivated through multiple examples and an empirical analysis of existing datasets, wherein this empirical analysis found that for two datasets 3.68% and 3.27% of samples would be impossible to classify through Liu's (Liu, 2015) original definition, due to ambiguity. Further by analysing the existing definition by Liu (2015), this thesis justifies parts of the definition through ambiguity, whereas in comparison the original justifications from Liu (2015) were motivated through application rather than a more theoretical ambiguity perspective.

- **The first reproduction study within TDSA.**
  Through the reproduction studies of two non-NN based methods (Vo et al., 2015; Wang et al., 2017a), neither report a factor that is significant in reproducing their results, further both of these factors, scaling features and the C-value of a Support Vector Machine (SVM) (Chang et al., 2011), are found across many datasets to be significant for both methods. Additionally it is found for NN based methods within TDSA that multiple runs can produce significantly different results, which is what Reimers et al. (2017) found for neural sequence labelling methods. The distribution of results created from the NN method through multiple runs is thus believed to be

[11] `https://jupyter.org/`.

[12] This is highlighted following what has been known as the #BenderRule (Bender, 2019), which re-iterated a point made in prior work (Bender, 2011), that not stating the language (normally English) the data has come from misleads the reader into thinking the work is language independent.

[13] Ambiguity is later defined, within section 2.4, as sentiment ambiguity.

the reason why prior works found it difficult to reproduce or replicate a particular NN TDSA method (Tang et al., 2016b).

- **The largest empirical evaluation of TDSA methods.**
  Due to our goal of reproducing diverse methods, in this case non-NN and NN methods, and applying them to a large range of different datasets, it was possible to test for generalisability. Finding that no one method was best or generalisable, but factors such as dataset size and sentiment class distribution determined whether it was best to use a NN or non-NN method. Further it was found for the NN methods that by evaluating on a larger and more diverse set of datasets than the original authors used, the novel NN methods that were created on some dataset were found to be no better on average than the original baseline method.

- **The creation of a new empirical evaluation methodology for TDSA.**
  Prior works' error splits are tested and formalised with the addition of new error splits and metrics created within this thesis. Through an extensive empirical evaluation, two of the existing error splits and one of the new error splits are not recommended to be used, due to them not measuring what they were hypothesised to measure. From the recommended error splits and metrics, one error split and the metrics have been created within this thesis to test for new phenomena within TDSA. These recommended error splits and metrics make up this new empirical evaluation methodology for TDSA. The new empirical evaluation methodology is then tested on multiple case studies whereas the prior work had only justified their hypothesis through small qualitative examples or using an error split that this analysis does not recommend. The case studies demonstrated the use of the empirical evaluation methodology. These analyses and experiments when combined create the largest and most extensive review of error analysis within TDSA to date.

- **An executable thesis.**
  As stated earlier this entire thesis is reproducible through the codebases and Jupyter notebooks that are attached to the relevant sections throughout the thesis. All the results that have been generated have used either the Bella[14] or target-extraction[15] packages created during the course of the PhD. The Bella package is more focused towards the non-NN methods whereas target-extraction is only focused on the NN methods. Both packages have extensive unit tests and Bella has an easy to use out of the box functionality through its model zoo.

1.4 Organisation of the Thesis

- **Chapter 2: Literature Review.**
  The majority of the literature review summarises the different levels of sentiment analysis starting with the most coarse grained (document level) and finishing with fine grained sentiment analysis. The review of coarse to fine grained sentiment analysis is one of, if not, the most extensive review of sentiment analysis, within the English language, showcasing the different granularities and how they link together.

[14] `https://github.com/apmoore1/Bella`
[15] `https://github.com/apmoore1/target-extraction`

Within the fine grained sentiment analysis review a new extended definition for fine grained sentiment analysis is stated. Further, the literature on TDSA has been reviewed showing the need for a reproducibility study. Further it motivates that error splits that do exist have not been rigorously analysed to better understand what they show and when they are useful. The review finishes with some further related topics to fine grained sentiment analysis, which motivate some of the future work within the conclusion chapter.

- **Chapter 3: Reproducibility and Generalisability of TDSA Methods.**
  The introduction motivates the need for performing both the reproducibility studies and generalisation experiments. The chapter provides a more extensive related work section detailing prior work within reproducibility more broadly and then concentrating on the most related work within sentiment analysis on both reproducibility and generalisability. The methods used in both the reproducibility and generalisability are described in detail. The reproduction studies are conducted, and the results are used to answer RQ 1. Lastly the generalisation experiments are performed whereby the results allow RQ 2 to be answered.

- **Chapter 4: Improving Experimental Methodology for TDSA.**
  This chapter contains an extensive review comparing and contrasting previous error splits within TDSA. Two new error splits are created to measure different phenomena, compared to the existing splits. All error splits are then reviewed on three English datasets, showcasing the difference between the datasets using the error splits. These error splits are then summarised describing what they do and what they hypothetically measure. The error splits are then tested across several TDSA methods and three English datasets to conclude if the error splits are measuring what was hypothesised. From these experiments, a new TDSA metric and its variants are created. The detailed review, analysis, and tests of the error splits in conjunction with the new metric allows for RQ 3 to be answered.

- **Chapter 5: Case Studies in Improving Experimental Methodology for TDSA.**
  The new experimental methodology for TDSA created within chapter 4 is then explored through several case studies. Each case study explores a new development whereby these new developments including position encoding, inter-aspect encoding, and transfer learning from a language model[16]. Each development is then applied to the methods used within chapter 4, when appropriate, and evaluated using the new experimental methodology. Within each case study the findings from the new experimental methodology are reviewed and where appropriate will be compared to the hypothesis that motivated the relevant development.

- **Chapter 6: Conclusion.**
  The conclusion summarises the thesis, revisits the research questions, and finishes with future work.

[16] Also known as contextualised word representations.

Chapter 2

Literature Review

2.1 Introduction

Sentiment analysis is an integral part of Natural Language Processing (NLP) and many parts of the NLP pipeline contribute to sentiment analysis methods. This pipeline is therefore the starting point of the review whereby the hierarchy of NLP tasks are described. The review then moves onto describing the many different layers of sentiment analysis, starting with coarse grained document sentiment analysis and ending with describing the many subtasks of fine grained sentiment analysis. This review not only synthesises the current relevant literature but also further extends the definition of fine grained sentiment analysis. In comparison to previous definitions of fine grained sentiment analysis this extended version removes any ambiguity that arises from the context, which is later defined in section 2.4 as sentiment ambiguity. Additionally the review finishes with further related topics on fine grained sentiment analysis, which is then later used to motivate some of the potential future work within the field in section 6.3.

2.2 Overview of Natural Language Processing

Natural Language Processing (NLP) as a field covers a wide range of computational tasks that all relate to the concept of creating computer-based models of either some aspect of language or more ambitiously all aspects of language. These models are then normally used to create explicit structure for the vast quantities of text that contains only implicit structure. NLP tasks can be categorised based on the linguistic property that the tasks are addressing e.g. a syntactic or semantic task. These linguistic properties and the related tasks are generally hierarchical, whereby knowing the output of a lower level task should help the higher level task. The low level syntactic tasks tend to be sequence labelling problems whereby each token[1] (Kaplan, 2005; Dridan et al., 2012) in the text (sequence) is assigned a label (Yannakoudakis et al., 2017), an example of this can be seen in figure 2.1 whereby each token is assigned a Part Of Speech (POS) tag. Examples of low level syntactic tasks are POS tagging (Church, 1988; Ling et al., 2015) and Chunking (Tjong Kim Sang et al., 2000). The low level syntactic information is usually used to guide the higher level syntactic tasks such as dependency (Nivre et al.,

[1] A token comes from tokenizing all words in the text using a tokenizer, of which this can cause some 'words' to be broken up further to produce more than one token per 'word'.

2007) and constituency parsing (Collins, 2003). As can be seen from the constituency and dependency trees within figure 2.1[2] and 2.2[3] the POS tags correlate largely with the constituency and dependency labels e.g. a noun (*NN*[4]) being part of a noun phrase (*NP*) and a noun (*NNS*[5] or *NN*) being the modifier in the nominal subject modifier (*NSUBJ*). This relation between low and high level task label/tag spaces has been shown to be effective to exploit when modelling the tasks (Hashimoto et al., 2017). The POS tags and constituency and dependency labels come from a limited tag or label set and the examples given here in figures 2.1 and 2.2 come from the Penn Treebank POS tag set (Taylor et al., 2003), syntactic tags of the Penn Treebank (Taylor et al., 2003), and Stanford typed dependencies (De Marneffe et al., 2008) respectively.

The relation between lower (POS tagging) and higher (supertagging) level syntactic tasks has been explicitly shown to be hierarchical within Søgaard et al. (2016) multi task learning[6] work. The relation between tasks extends beyond the categories where syntactic information is useful for semantic based tasks (Hashimoto et al., 2017), for instance within text classification. The example sentence within the figures is a neutral sentence as it contains an equal amount of positive and negative sentiment[7]. To know the sentiment of the sentence it requires knowing that words such as *great* are positive words, this type of knowledge is generally referred as semantic knowledge. Further for the example sentence knowing that the words *was n't* modifies the meaning of the word *great* requires syntactic (from the dependency tree, figure 2.2) and semantic information. Similar to the syntactic tasks, semantic tasks also have a hierarchical structure where some tasks require less language understanding than others (Sanh et al., 2019). For a more comprehensive overview of syntactic and semantic tasks in NLP and how they relate, see Chapter 6 and 7 in Goldberg (2017). In recent years it has been shown that utilising Neural Networks (NN) that have been initially trained on a high level NLP task such as Masked Language Modelling (MLM) (Devlin et al., 2019) can be useful for the whole NLP pipeline (syntactic to semantic tasks) (Tenney et al., 2019). Thus finally showing how the tasks are hierarchical in nature. This brief primer into NLP does not touch on any topics that utilise NLP with any other modality, such as images for image captioning (Karpathy et al., 2015), audio for sentiment analysis (Raaijmakers et al., 2008), and many more, but these areas are out of scope for this thesis.

[2]Constituency tree demo URL `https://demo.allennlp.org/constituency-parsing`

[3]Dependency tree demo URL `https://demo.allennlp.org/dependency-parsing`

[4]Singular noun.

[5]Plural noun.

[6]For an introduction into multi task learning and how it relates to transfer learning (which will be mentioned later in this section) see (Ruder, 2019) chapter 3 and 3.2.

[7]If the sentence is put through Stanford CoreNLP 3.9.2 it would also classify the sentence as neutral. URL to Stanford CoreNLP 3.9.2 `https://corenlp.run/`

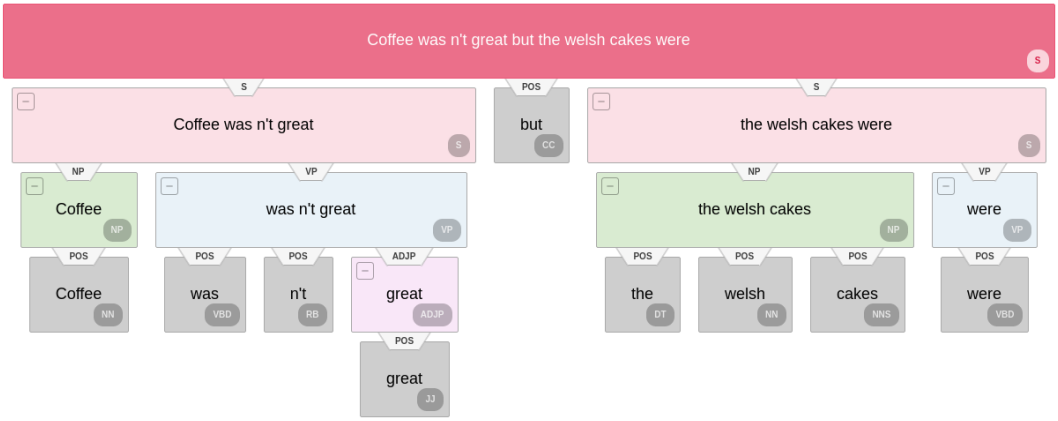


Figure 2.1: Constituency tree for the sentence 'Coffee wasn't great but the welsh cakes were'. The labels in each box represents the constituency label for that span, the labels for the leaf nodes are the POS tags for those words. The tree was created using the AllenNLP demo, which used Joshi et al. (2018) model.

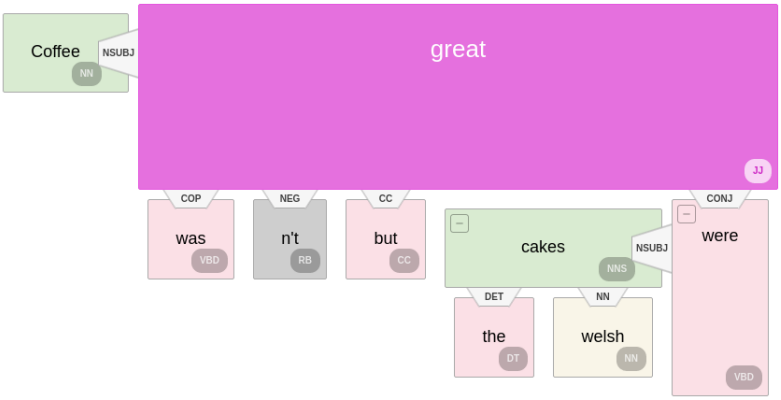


Figure 2.2: Dependency tree for the sentence 'Coffee wasn't great but the welsh cakes were'. The labels within the arcs are dependency labels, and the labels within each box represents the POS tag for the given word. An arc from *word A* to *word B* indicates that *B* is modifying *A*, where *A* is the head word. For example *great* is the head word of *coffee*. The tree was created using the AllenNLP demo, which used Dozat et al. (2017) model.

2.3 Sentiment Analysis

Sentiment analysis can be seen as a more general topic that contains multiple different sub-tasks. These tasks tend to be related and often have different assumptions. In this section, various coarse grained sentiment tasks within sentiment analysis will be discussed starting with the most coarse, document level, and ending with aspect based. During the discussion some important concepts within NLP will also be introduced such as different neural network methods. Throughout the section the different tasks will be shown how they link to each other. The summary of coarse grained sentiment analysis is a primer to the fine grained sentiment analysis review in section 2.4, which contains the main topic of this thesis Target Dependent Sentiment Analysis (TDSA).

The many methods that have been used to tackle TDSA and all NLP problems fall into three different categories; supervised, unsupervised, and semi-supervised. Supervised methods require labelled data, in the case of sentiment analysis this would be sentiment labels attached to their relevant text (Pang et al., 2002). Unsupervised methods on the other hand only require unlabelled data, which in the sentiment analysis case is just the text. Unsupervised methods within sentiment analysis are normally based around sentiment lexicons (lists of words that have an attached sentiment) (Hu et al., 2004a) and sometimes a combination of rules (Hutto et al., 2014). Thus unsupervised methods tend to require some prior knowledge. Finally, semi-supervised is a combination of the two, learning from both the labelled data and extra unlabelled data (Zhu, 2005). For a more complete overview of the differences between supervised, unsupervised and semi-supervised, see Weston (2007).

2.3.1 Document Sentiment Analysis

The most common sentiment analysis task is that of document level. The task here is given a document which is made up of multiple sentences, to predict the sentiment with the assumption that the document is about one topic (Nasukawa et al., 2003). An example sample for this task can be seen in example 1 where the sentiment of the whole document/review is assumed to be about the movie that is being reviewed. The first to apply a supervised machine learning algorithm to this problem was Pang et al. (2002), where they applied several Machine Learning (ML) classifiers with Bag Of Words (BOW) as features to a new movie review dataset. This line of research of applying ML was further extended by Pang et al. (2004) who found that they could reduce the number of sentences by removing objective sentences in the document without significantly impacting, and in some cases improving, the overall accuracy of the classifier. Mullen et al. (2004) explored incorporating more semantic features into the BOW models such as the average sentiment values based on the unsupervised techniques of Turney (2002). This approach of adding more semantic information into BOW models was further explored by Whitelaw et al. (2005) who created lexicon feature sets based around appraisal groups. Even though the use of semantic information had improved results (Whitelaw et al., 2005), the strong baseline performance of just using n-gram BOW features was further investigated by Martineau et al. (2009) who showed that incorporating the class into the TF-IDF weighting mechanism (Jones, 1972) to create Delta TF-IDF significantly improved results. However Delta TF-IDF was only created to work with two classes[8] and not shown to generalise to n classes. However a future study by Paltoglou et al. (2010) showed that further performance gains can be made to TF-IDF based systems by using more enhanced weighting systems like BM25 (Robertson et al., 1995).

[8] In the sentiment case this is the positive and negative classes.

a couple of criminals (mario van peebles and loretta devine) move into a rich family's house in hopes of conning them out of their jewels . however , someone else steals the jewels before they are able to get to them . writer mario van peebles delivers a clever script with several unexpected plot twists , but director mario van peebles undermines his own high points with haphazard camera work , editing and pacing . it felt as though the film should have been wrapping up at the hour mark , but alas there was still 35 more minutes to go . daniel baldwin (i can't believe i'm about to type this) gives the best performance in the film , outshining the other talented members of the cast .

Example 1: Negative document level sentiment example. Document ID *cv435_24355* taken from Pang et al. (2002) sentiment dataset.

The supervised approaches that have been mentioned so far use a BOW approach, of which this form of vector representation is limited in what it can represent. BOW approaches that use n-gram word features can only learn what a word means within that n window. For instance take n to be 1 and 2[9] it would understand terms such as 'very good' and 'good' where both would be associated with positive sentiment, however if the statement was 'not very good' then it would not capture the full sentiment as it would require all three words to know it is negated. One approach would be to have a very large value for n, but this would create a very sparse vector representation which would not generalise well (Le et al., 2014). Thus the move away from BOW sparse vector representations to dense word representations was shown to be promising for sentiment analysis in Maas et al. (2011) work. However, this work only found better performance than BOW when they combined the dense vectors with the BOW sparse vector. This first step into dense representation did show some promise as the representation can be learnt from unlabelled data, where they found that results increased when more unlabelled data was used. These dense representations are very similar to what a traditional BOW model learns through its weights within the model (Goldberg, 2017)[10]. The benefit of using the dense representations on the downstream task (sentiment analysis) is that a good representation of the vectors can be learnt from another unsupervised task from unlabelled data first[11]. Thus allowing the model to have prior knowledge of what words mean (semantically and syntactically (Mikolov et al., 2013b)) is encoded into the vector representation before training the model, unlike the BOW representation which contains no prior knowledge.

Le et al. (2014) showed for the first time how dense vector representations using a NN could surpass BOW representation for which Wang et al. (2012) set a high baseline at the time for a BOW method. Le et al. (2014) created dense document/paragraph vector representations that in comparison to the prior word level versions (Maas et al., 2011) could encode document size texts without averaging by learning to predict the next word within a small context window from the document, thus each document vector would be different unlike the word representations[12]. Johnson et al. (2015) showed that without any additional unlabelled data unlike Le et al. (2014) a Convolution NN (CNN) can outperform the BOW approach. The CNN can be seen as a NN approach to a

[9] i.e. uni-gram and bi-gram features.
[10] See section 2.5.
[11] The task can also be supervised, but would require labelled data.
[12] Unless two or more documents are identical.

BOW model whereby the CNN has a set of user defined window sizes of which these window sizes are analogous to n-grams in a BOW model. The CNN differs from the BOW model as it can generalise to unknown n-gram sequences, as it learns how to combine representation from multiple word representations to create the n-gram representation. In comparison for the BOW model it learns what that entire n-gram means and disregards similarities between n-grams based on the words within the n-gram. Furthermore the CNN model does not have the sparsity problem that a BOW model has thus the window size (n-gram in BOW case) can be large; e.g. Johnson et al. (2015) found that having a window size of 2 and 3 performed best[13]. For a more complete overview of CNNs, the reader is directed to chapter 13 of Goldberg (2017).

None of the above methods, including the NN approach, take into account the word order of the whole document. One family of NN that explicitly encodes the whole sequences of text in order is the Recurrent NN (RNN) (Rumelhart et al., 1985). The RNN has several popular variants; Long Short Term Memory (LSTM) (Hochreiter et al., 1997) and the Gated Recurrent Unit (GRU) (Cho et al., 2014). Dai et al. (2015) showed practically that LSTMs can be used for long sequence classification tasks such as document sentiment classification. Xu et al. (2016) created the Cached LSTM to better encode information from large sequences of text, like documents, and showed that it can outperform the LSTM on document sentiment analysis. Hierarchical (Zhang et al., 2015b) and dilated (Strubell et al., 2017) CNN approaches have been created which can capture large contexts, e.g. sentences rather n-grams, where they have been shown to be successful in sentiment analysis (Conneau et al., 2017). Finally and more recently, the transformer NN (Vaswani et al., 2017) approach has been applied which does not preserve word order (like the RNN or CNN) but rather treats the text more like a tree structure through attention mechanisms, whereby each word learns to contextualise itself within the text (can be the whole text). The transformer success can be best seen through BERT (Devlin et al., 2019) where the architecture can be applied to a task like document sentiment analysis (Sun et al., 2019b)[14].

The majority of these more recent NN approaches have been tested on much larger sentiment datasets (100Ks of documents) compared to the earlier work (2-25K documents). It has also been shown that for some of these NN approaches to work well they require extra data (Dai et al., 2015). However it was found in Dai et al. (2015) that these NN approaches can make great use of unlabelled data through a language modelling objective. This technique of training on one or more datasets and/or tasks before then applying the model to the end task (in this case document sentiment analysis) is defined as transfer learning in this thesis (Ruder, 2019)[15]. Both Howard et al. (2018) and Sun et al. (2019b) found that by pretraining (a type of transfer learning) on the unlabelled data can have large performance gains, making the models more sample efficient with respect to labelled/annotated data. All of the neural methods within this and the last paragraph take into account large contexts of the document, if not the whole document, whereas none of the previous methods could effectively do so. This allows the methods to overcome one the main challenges in document sentiment analysis that both Turney (2002) and Pang et al. (2002) found where the sentiment of the whole document is not

[13]Le et al. (2014) has a good summary of the drawbacks of BOW vector representations.

[14]For a good comparison of RNN, CNN, and transformer based models with respect to computational cost see section 4 of Vaswani et al. (2017).

[15]Chapter 3.

the "sum of the parts" (Turney, 2002).

The approaches taken so far treat the document as one large object to encode using some form of supervised method. However there have been methods proposed that suggest it is better to break the document up into smaller linguistic units to encode first into some form of intermediate representation, which is then later combined to generate a sentiment value for the whole document. Bhatia et al. (2015) used Rhetorical Structure Theory (RST) (Mann, 1984) to create a discourse structure for the document, which can be represented as a dependency-based discourse tree where the nodes are represented by Elementary Discourse Units (EDUs). They found that weighting the outputs of different supervised and unsupervised methods applied to EDUs in the tree based on their depth within the tree improved results. Yang et al. (2016) used sentences to represent the whole document and weighted these sentences and the words within each sentence using two supervised attention mechanisms. Thus this method unlike Bhatia et al. (2015) learnt which sentences and words within those sentences were important to the document's sentiment rather than having a predefined weighting mechanism.

Pang et al. (2002) created the first English dataset of movie reviews (1.4K documents), which was later revised and increased in Pang et al. (2004) (2K documents), and then increased a lot further by Maas et al. (2011) (25K documents). The prior datasets all contained only two classes, positive and negative, and these were based on the star rating that the movie review was given by the user. Zhang et al. (2015b) created four much larger datasets (600K – 4M documents) that originated from Yelp[16] and Amazon reviews (McAuley et al., 2015) which contain between two and five classes where the classes are based around the user's star rating. This list of English datasets is not supposed to be exhaustive but is given as reference to popular and widely used datasets from the past and current literature.

In this sub-section, document sentiment analysis has been covered with respect to supervised methods and the associated popular English datasets. From the literature it is clear to see that the SOTA are NN based methods that require some form of transfer learning (Yang et al., 2019). All of the methods also clearly point to one of the main challenges in document sentiment analysis which is how best to contextualise the whole document. In the early methods, such as BOW, the methods could only create local contexts through n-grams. This was overcome with NN approaches such as the LSTMs, hierarchical CNNs, and transformers being able to capture the global context of all tokens in the document. Finally within this sub-section different NN approaches have been explained as well as defining the concept of transfer learning.

2.3.2 Sentence Sentiment Analysis

Sentence level sentiment analysis is very similar to document level, whereby the only difference is the length of the text to be processed. Due to the length difference, the task is somewhat conceptually easier as a sentence is less likely to have multiple conflicting sentiments, thus the overall sentiment of the sentence is easier to predict. Many different approaches have been used for sentence-based analysis: BOW (Wang et al., 2012), CNN (Kim, 2014; Kalchbrenner et al., 2014), LSTM (Brahma, 2018), and BERT (Devlin et al., 2019). However, similar to document level, "the sentiment of a sentence is not merely

[16] https://www.yelp.com/dataset

the sum of the polarity of the words and phrases found in the text, but rather depends on a number of compositional phenomena that act on indicators of polarity" (Barnes et al., 2021). This problem can be best seen in figure 2.3[17], where it can be seen that even though the sentence has multiple positive words the sentiment of the sentence is dominated by the scope of the negation. Due to this, many approaches have been taken to better model the sentiment phrases within the sentence using BOW with compositional semantics (Choi et al., 2008), Conditional Random Field (CRF) (Nakagawa et al., 2010), Recursive NN (RCNN) (Socher et al., 2012), deep RCNN (Irsoy et al., 2014), Tree-LSTM (Tai et al., 2015), Graph NN (GNN) (Zhang et al., 2019c), and multi task learning whereby negation scope and cue detection are the auxiliary task (Barnes et al., 2021).

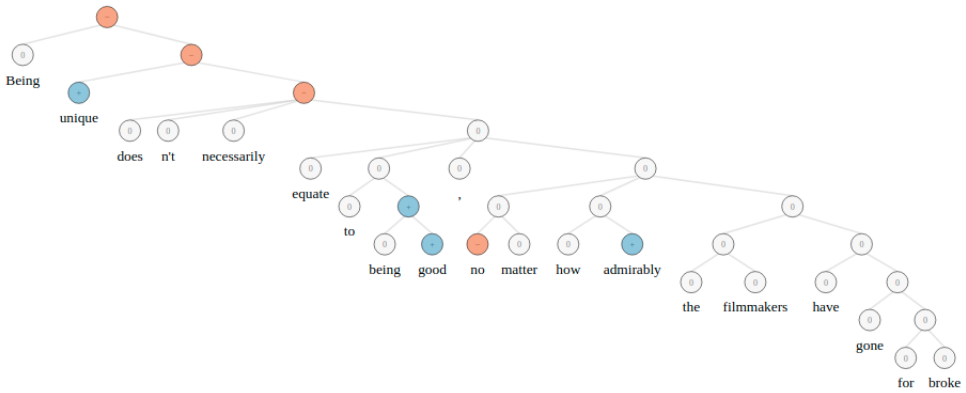


Figure 2.3: Phrase and overall sentiment from Socher et al. (2013) model, where red, white, and blue represent negative, neutral, and positive sentiment respectively. The sentence in the figure is 'Being unique doesn't necessarily equate to being good, no matter how admirably the filmmakers have gone for broke'.

Socher et al. (2013) took modelling phrases within sentences further by creating a dataset where each phrase from the Pang et al. (2005) movie dataset was manually annotated with a sentiment value, which can be seen in figure 2.3 whereby the model output shown is how the dataset is annotated. Thus this dataset allows models to better capture the compositional phenomena more explicitly by using the phrase level annotation. In a similar trend, Yang et al. (2014) found using inter- and intra-sentence discourse features to be useful, showing that sentence level classification can be dependent on surrounding sentences. Similarly McDonald et al. (2007) found that jointly modelling the sentence and document classification tasks helps improve both. More recently, Angelidis et al. (2018) found that by framing document classification as a multiple instance learning (MIL) (Dietterich et al., 1997) problem they were able to create a sentence (EDU) classifier using only the document labels, this outperformed a fully supervised sentence (EDU) level classifier. This MIL method was also shown to be useful when applied to the food health domain (Karamanolakis et al., 2019).

This review has shown that sentence level sentiment requires both understanding the complex structure within the sentence as well as the more global content of the document. However, even though the literature shows the use of explicitly taking into

[17]The sentence was chosen from Barnes et al. (2021) and the figure was generated using the live demo found at this URL `http://nlp.stanford.edu:8080/sentiment/rntnDemo.html`.

account phrases and document level information, current SOTA uses the same techniques as that of document level (Yang et al., 2019), utilising transfer learning mainly from a language modelling task. Even so, Barnes et al. (2019) have annotated the errors coming from SOTA models into eighteen different categories finding that they perform badly on sentences containing non-standard spellings, idioms, and world knowledge, to name a few. This shows that sentence level sentiment analysis still has plenty of error cases to solve through better incorporating linguistic and world knowledge into new and/or existing methods. To clarify, all of the research mentioned on sentence level sentiment analysis was applied and thus to some extent developed for English.

2.3.3 Aspect Based Sentiment Analysis

Document and sentence level sentiment analysis both assume that they are discussing one topic (Liu, 2015)[18], for instance if the document (sentence) comes from a review (headline of a review) of the movie, *The Avengers*, the topic that the sentiment is assumed to be about is the movie, *The Avengers*. However, both documents and sentences can contain sentiments on multiple different topics not just the main overall topic of the entire review. Aspect Based Sentiment Analysis (ABSA) attempts to overcome this problem, instead of predicting one sentiment for a document or a sentence it predicts multiple sentiments conditioned on multiple different predefined aspects/topics. ABSA can be performed at different linguistic granularities, typically either document or sentence level. Example 2 is a Tripadvisor review taken from Wang et al. (2010) with seven aspects and their respective sentiments[19], as well as the overall sentiment of the hotel. From this document level ABSA example it can be seen that these aspects are latent, that is the aspect itself does not necessarily occur in the text, this is shown in example 2 where the *service* is negative as the "Hotel staff speak zero English... The process at the hotel is a bit confusing... staff weren't overly friendly.". This example would be typically used as a training example for document level ABSA. In this subsection document and sentence level ABSA will be described along with advances in each area and popular datasets.

Good and clean but no English spoken The good: The hotel is in a great location and withing walking distance to the Forbidden City and some other sights in this hisrotic district. The rooms were comfortable and clean, really good value.The bad: Hotel staff speak zero English. Breakfast is only Chinese breakfast. The process at the hotel is a bit confusing (make sure you keep those pink receipts they give when you pay, you need them to check out!) and staff weren't overly friendly. Internet is expensive and slow.

Example 2: Example of document level aspect sentiment analysis. The aspects and their receptive sentiments are: *service* (1), *business service* (2), *cleanliness* (3), *check in / front desk* (2), *value* (4), *rooms* (3), and *location* (4). The sentiments were on a scale of 1-5 and the overall sentiment for the review was 2. This was taken from review id *447367* from the trip advisor review dataset of Wang et al. (2010).

[18] Page 47-48.

[19] They were actually ratings rather than sentiments, but the ratings are used as approximations for sentiment.

2.3.3.1 Document-level ABSA

Snyder et al. (2007) treated the task of document ABSA as a supervised ranking problem, where each aspect is ranked based on a BOW feature vector. They showed the benefit of modelling dependencies between aspects. They found that 38% of their restaurant review training data contained the same sentiment for all aspects in their respective reviews. Thus they added an agreement function within their model which predicts whether or not all aspects for the review should contain the same sentiment which significantly improved performance. In comparison Wang et al. (2010) created the Latent Rating Regression model, which instead of learning from the aspect sentiments can train the model just from the overall sentiment. This therefore reduces the requirement of more fine grained sentiment training data, but the model did require a set of seed words that represented the aspects e.g. for the aspect *room* a set of key words would be *room*, *suite*, *view*, and *bed*.

More recently, supervised Neural Network (NN) approaches have been the most popular and successful approach to document ABSA. The first to use a NN for document ABSA was Lei et al. (2016), whose main aim was to create rationals/explanations for the predictions given only the aspect sentiment labels for supervision. Yin et al. (2017) showed the importance of biasing the attention mechanism within a hierarchical NN (Yang et al., 2016) towards the aspect of interest. The attention mechanism used a set of keywords to define each aspect, from these keywords the attention mechanism would use a memory network (Weston et al., 2015) to learn how to best describe the aspect so that the model focused on the most important words and sentences in the document. Yin et al. (2017) benchmarked their approach across numerous neural and non-neural approaches. Li et al. (2018a) used a similar hierarchical NN as Yin et al. (2017) and found significant performance gains when incorporating both user[20] and/or the overall document sentiment information into the attention network. They found that the document sentiment information is useful as the related aspect sentiments are correlated. Further the user information allows the model to better capture textual and sentiment similarities at the aspect level, e.g., a user tends to have similar sentiment scores for aspects across documents and tends to describe aspects in a similar manner across documents. Finally, the most recent and successful approach used a hierarchical neural Reinforcement Learning (RL) (Williams, 1992) method (Wang et al., 2019)[21], whereby instead of splitting the document into sentences they used Elementary Discourse Units (EDUs). They used the RL approach to first find aspect relevant EDUs and then within the EDU the relevant aspect sentiment words. They found that if they had used sentences instead of EDUs[22] then the performance would have decreased by 2.44% on average, of which they believe this is due to 90% of EDUs only containing one sentiment (Bayoudhi et al., 2015). Through error analysis they found that their method performed poorly when negation is used or a comparison is made, however they did not quantify the number of times these errors where made.

There are three main datasets to evaluate document ABSA, Tripadvisor (Wang et al., 2010)[23], BeerAdvocate (McAuley et al., 2012), and TripUser (Li et al., 2018a). Both

[20] On the dataset that contained user information.

[21] They called document ABSA, Document-level Aspect Sentiment Classification (DASC).

[22] They call EDUs clauses.

[23] This is sometimes called TripDMS.

the Tripadvisor and BeerAdvocate datasets since have been reprocessed by Yin et al. (2017) so that aspect sentiments per document are less correlated[24][25]. This technique of reducing the correlation of the aspect sentiments per document was suggested by Lei et al. (2016), to ensure that the model does not get "confused". It has not been shown whether training models on datasets that have less correlation between aspect sentiments per document produce better models, further Snyder et al. (2007) actually exploited this correlation in their modelling. The Tripadvisor and TripUser are both hotel reviews from the Tripadvisor website[26] and contain the same seven aspects, but the TripUser dataset also contains user information unlike Tripadvisor. The BeerAdvocate dataset comes from a beer review website Beeradvocate[27] and contains four aspects.

The datasets mentioned so far are all the datasets that have been used[28] in the prior work stated in this thesis so far. However these datasets are not expertly annotated data, rather the data has been scraped from their representative websites where the reviews were 'annotated' by many different users. One of the few datasets that has been annotated by experts is the SemEval 2016 task 5 subtask 2 dataset (Pontiki et al., 2016), which contains seven datasets in five different languages and varies across three domains[29]. The only methods that have been applied to these datasets are those that entered the SemEval competition.

2.3.3.2 Sentence ABSA

Sentence ABSA unlike document ABSA tends to contain far fewer aspects within its text and unlike document it is rare for a sentence to have all aspects, which is not the case in document ABSA (Snyder et al., 2007; Wang et al., 2010). All document ABSA methods could be applied to sentence ABSA with minimal changes, however the vast majority of them have been designed for longer texts, for instance the hierarchical NN methods (Yin et al., 2017; Li et al., 2018a; Wang et al., 2019). To further iterate this point on a sentence containing few aspects per sentence, examples 3 and 4 show two different sentences from the widely used SemEval 2014 task 4 subtask 4 restaurant review dataset (Pontiki et al., 2014), containing one and two aspects respectively. Further, the statistics from the SemEval 2014 restaurant training dataset state that on average each sentence will only have 1.22 aspects[30] (Pontiki et al., 2014). In comparison to document, sentence ABSA is less likely to have an *overall* sentiment[31], of which some document approaches have made use (Wang et al., 2010; Li et al., 2018a). Lastly, document ABSA

[24]The standard train, development, and test splits for the Tripadvisor and BeerAdvocate datasets can be found here `https://github.com/HKUST-KnowComp/DMSC`.

[25]The less correlated method comes from Lei et al. (2016, §5.1). They train a linear regression model to predict one of the aspect's sentiment based on all the other aspects sentiments, they then pick the documents that have the largest error until the aspect sentiment correlation in the documents goes beyond a certain threshold.

[26]`https://www.tripadvisor.co.uk/`

[27]`https://www.beeradvocate.com/`

[28]Or derivatives of.

[29]Restaurant, laptop, and hotel reviews.

[30]This was calculated based on the training datasets containing 3041 sentences of which in total there are 3713 aspects within that dataset.

[31]*Overall* sentiment here refers to sentence or document level sentiment, rather than an *overall/general* aspect sentiment e.g. the aspect *RESTAURANT#GENERAL* in the SemEval 2015 task 12 restaurant dataset (Pontiki et al., 2015).

in effect summarises the sentiment information for each aspect that has been captured throughout the document, which can make document sentiment analysis a more difficult task if there are lots of contradicting sentiments for one aspect (Pontiki et al., 2016).

Overall I would recommend it and go back again.

Example 3: Example of sentence level aspect sentiment analysis. Contains one aspect *anecdotes/miscellaneous* with positive sentiment. This was taken from sentence id *2609* from the trail restaurant dataset of Pontiki et al. (2014).

Even though its good seafood, the prices are too high.

Example 4: Example of sentence level aspect sentiment analysis. Contains two aspects *food* and *price* with positive and negative sentiment respectively. This was taken from sentence id *3440* from the trail restaurant dataset of Pontiki et al. (2014).

Sentence ABSA was popularised by task 4 in SemEval 2014 (Pontiki et al., 2014) where 20 teams created various methods. The winner, Kiritchenko et al. (2014), used a Support Vector Machine (SVM) (Chang et al., 2011) with different BOW features including ngrams, POS tags, and various in and out of domain sentiment lexicon features[32]. To better capture the aspect specific sentiment, they utilised a domain adaptation technique (Daumé III, 2007) such that each aspect category had its own BOW weight vector that was learnt at the same time as the general BOW weight vector, so that aspect specific and general features can be learnt separately. SemEval repeated a similar task for two more years, 2015 and 2016, where the winners all use a similar BOW approach (Saias, 2015; Brun et al., 2016; Kumar et al., 2016).

NN approaches for this task are desirable due to them requiring fewer linguistic resources, and the ease of transferring the approach to multiple languages (Ruder et al., 2016b). Ruder et al. (2016b) applied a CNN to the task outperforming many of the aforementioned BOW approaches on multiple languages. In later work Ruder et al. (2016a) showed that taking into account the surrounding sentences using an hierarchical LSTM further improved results. Wang et al. (2016b) found that by adding attention to a sentence level LSTM improved the model by allowing it to better capture relevant aspect specific words within the sentence. Follow on work (Bao et al., 2019) found that regularising the attention network using sentiment lexicons and/or attention sparsity improved the robustness of the model. Wang et al. (2018) utilised a hierarchical NN similar to Yin et al. (2017) (document ABSA) whereby the hierarchy of the sentence was based around EDUs within the sentence, and the words within those EDUs. This was proposed on the premise that EDUs often discuss one aspect, thus making the task for the NN easier, as the model could ignore EDUs that were not discussing the relevant aspect. They found this hierarchical approach to outperform the flattened version. Current SOTA approaches utilise transfer learning from Bi-directional Language Models (BiLM) (Sun et al., 2019a; Jiang et al., 2019) (also known as Contextualised Word Representations (CWR) which is what they will be called from now on).

[32]Sentiment lexicon is a collection of different groups of words where each group is associated to a specific sentiment. This definition follows that of Mohammad et al. (2010) for their emotion lexicon definition if you substitute emotion for sentiment.

Compared to the previous approaches Kaljahi et al. (2018) explored whether adding the aspect's sentiment expressions to a model would improve results. The sentiment expression which is "part of the sentence which conveys the sentiment towards a certain aspect" (Kaljahi et al., 2018) was added to the English SemEval 2016 laptop and restaurant datasets. Example 5 demonstrates what an aspect's sentiment expression is. They found that adding the sentiment expressions into the NN model improved results greatly, however in this setup at test/inference time the model would require the aspect's sentiment expression. To overcome this they incorporated the sentiment expressions in a multi task setup, but this led to the model performing as well as a model that did not require sentiment expressions. This line of research shows promise as it demonstrates that using aspect sentiment expressions can improve results but incorporating this information into a model that does not require it at test time is difficult. Further aspect sentiment expressions, if they could be predicted at the same time as the sentiment, could create some form of explanation to the sentiment prediction, making the black box NN methods more explainable, which is similar to what Lei et al. (2016) did in document ABSA.

However, go for the ambience, and ***consider the food just a companion*** *for a trip across the world!*

Example 5: Aspect sentiment expression example, where the sentiment expression is in **bold** for the related aspect **food#quality**. This was taken from the SemEval 2016 Restaurant dataset (Pontiki et al., 2016) with the additional sentiment expression labelled by Kaljahi et al. (2018).

Popular sentence ABSA datasets for English are the SemEval 2014[33] (Pontiki et al., 2014), 2015 (Pontiki et al., 2015), and 2016 (Pontiki et al., 2016) datasets for the laptop, restaurant, and hotel review domain[34]. A large difference between the 2015 and 2016 datasets compared to the 2014 was the sentiment of an aspect may require a larger context than just the sentence the aspect appeared in, hence why the whole review was given as context. The 2016 SemEval dataset also expanded the number of languages from just English to six more languages[35]. Also, as stated in the last paragraph, Kaljahi et al. (2018) annotated the English SemEval 2016 laptop and restaurant dataset with sentiment expressions. There has also been a challenge dataset, Multi-Aspect Multi-Sentiment (MAMS) (Jiang et al., 2019), which is within the restaurant review domain[36], unlike the other datasets it ensures that each sentence contains at least two aspects and at least two different sentiments per sentence.

Sentence level ABSA has also been known as topic sentiment analysis within the Twitter sentiment analysis community. Topic sentiment analysis has been run as a competition three times at SemEval (Rosenthal et al., 2015; Nakov et al., 2016; Rosenthal et al., 2017), each year creating a new larger dataset for English and in the final year (2017) creating an Arabic Twitter dataset as well. In their first year of running the competition (2015) they also ran a task (D) which they described as "sentiment towards a topic in a set of tweets" which can be viewed as a document ABSA task. These

[33]The restaurant ABSA dataset partially came from Ganu et al. (2009).

[34]The 2014 datasets ABSA annotation was only provided for the restaurant domain.

[35]The six languages are; Dutch, French, Russian, Spanish, Arabic, and Chinese.

[36]This dataset was created from the same source of restaurant reviews as the SemEval 2014, 2015, and 2016 restaurant dataset, which was the Citysearch New York dataset by Ganu et al. (2009).

datasets are also annotated for Tweet/sentence sentiment as well as the topic, the 2017 dataset contains user information, and the 2015 dataset is also annotated with sentiment expressions.

From this section on ABSA prior work it is clear that there is plenty of future work that can be explored. Creating more robust models is important, especially for real world applications. Bao et al. (2019) explored this problem, however they only evaluated on one dataset, thus expanding this evaluation for more datasets and across a spectrum of low to high resource settings would be of use to better understand how robust their method is. An unexplored area is the link between sentence and document ABSA whereby one would assume that document ABSA should improve the performance of sentence and vice versa, of which this can be empirically tested using the SemEval 2015 (Pontiki et al., 2015) and 2016 (Pontiki et al., 2016) review datasets.

2.4 Fine Grained Sentiment Analysis

Compared to the previous sections, this as the name suggests, is the most detailed granularity of sentiment analysis. Fine grained sentiment analysis is made up of many different tasks where the results of these tasks when combined creates the information required to fully understand a sentiment in context. The literature has been very clear on the following tasks being at the core of understanding sentiment in context (Wiebe, 1994; Kim et al., 2004; Ding et al., 2008; Liu, 2015):

1. Identifying the holder of the sentiment.

2. Identifying the target/object of the sentiment within the text[37].

3. The sentiment of the holder towards the target.

The information from these three tasks creates a triplet of information (sentiment holder, target, sentiment) to define the whole fine grained sentiment task. To better understand this, example 6[38] contains five triplets; (author, food, negative), (Jonathan, bara brith, positive), (He, leeks, neutral), (June, Welsh fruit cake, negative), and (June, chocolate cake, positive).

(1) The ***food*** *at the store was horrible, but Jonathan thought the* ***bara brith*** *was fantastic.(2) He did think the* ***leeks*** *were so so. However June found the* ***Welsh fruit cake*** *to be tasteless but the* ***chocolate cake*** *to be great.*

Example 6: Made up fine grained sentiment analysis example written on the 24th May. For reference bara brith is a kind of fruit cake that was created in Wales.

The triplet of information, which to some degree, was first defined within Wiebe

[37] In Ding et al. (2008) they use the word feature to mean target. In Kim et al. (2004) they use topic to mean target.

[38] In all examples the target will be in bold, the sentiment words within the text in the colour of their sentiment, and the holder underlined. The sentiment colours are pink for negative, grey for neutral, and blue for positive.

(1994) when stating the components of a private state[39] has been expanded by Liu (2015) who has created the clearest and most detailed definition so far and builds upon the previous work (Wiebe, 1994; Kim et al., 2004; Ding et al., 2008). Liu's (Liu, 2015) basic quadruple definition[40] is the same as the triplet so far with the added information of time. The importance of time within the definition is argued as sentiment can change over time, which is of importance for real world applications as one may want to know how a holder's[41] sentiment changes.

This quadruple definition is then expanded to a more detailed quintuple definition[42], with the target of the sentiment replaced with entity and aspect of the sentiment; (sentiment holder, aspect, entity, sentiment, time). Both the entity and aspect are latent in that they do not have to link to any part of the text unlike the target within the triplet. Further, the aspect and entity are linked via a hierarchical structure as in the aspect would be either an attribute or part-of the entity. The motivation behind removing the target and replacing it with the aspect-entity pairing is so that targets can be better grouped together. For instance, in example 6 even though both June and Jonathan are discussing the same food produce (bara brith) due to the different ways of stating it, grouping would have to occur to know they are discussing the same thing. The benefit of this aspect grouping is that from an application perspective an end user can get an overall trend of a sentiment towards aspects and entities of interest rather than sentiment towards targets, given that there could be exponentially more targets compared to aspect-entity pairs. To make the quintuple definition more concrete, the quintuples for example 6 are; (author, GENERAL, FOOD, negative, 24$^{\text{th}}$ May), (Jonathan, CAKE, FOOD, positive, 24$^{\text{th}}$ May), (He, VEGETABLES, FOOD, neutral, 24$^{\text{th}}$ May), (June, CAKE, FOOD, negative, 24$^{\text{th}}$ May), and (June, CAKE, FOOD, positive, 24$^{\text{th}}$ May).

The problem with the quintuple definition is that it is lacking in target information. Even though the target might be too fine grained for some applications, it can be useful for training models as the position of the target is useful for inferring the sentiment (Gu et al., 2018). More so there could be a case where a text contains two contradicting sentiments for the same aspect-entity pair, holder, and time. This can be seen in example 6 whereby June is negative about the Welsh fruit cake but positive about the chocolate cake. This would be confusing for training a machine learning model as the same text contains contradicting sentiments for the same aspect, entity, holder, and time. This problem of having two contradicting sentiments for the same information and text is defined in this thesis as sentiment ambiguity, in this case the information is aspect-entity pair, holder, and time. To avoid this sentiment ambiguity problem one could simply create more detailed aspect or entities which to some degree is suggested in Liu (2015)[43] or as proposed in this thesis, add the target to the quintuple to create the hextuple as in definition 1. To make this concrete, the following are hextuples for example 6; (author, food, GENERAL, FOOD, negative, 24$^{\text{th}}$ May), (Jonathan, bara brith, CAKE,

[39]Page number 235 of Wiebe (1994), defines experiencer, attitude, and object as the components of a private state. Experiencer, attitude, and object can be mapped to holder, target, and sentiment within the triplet of information definition.

[40]Definition 2.1 in Liu (2015).

[41]A more concrete example of a holder could be a politician, whereby the application may want to see when/if they changed their sentiment towards a bill/law.

[42]Definition 2.7 Liu (2015).

[43]On page 22-23 of Liu (2015) they state that if the user wants more detailed information then more entities would need to be created, and they use a printer and its ink as an example.

FOOD, positive, no time given), (He, leeks, VEGETABLES, FOOD, neutral, 24th May), (June, Welsh fruit cake, CAKE, FOOD, negative, 24th May), and (June, chocolate cake, CAKE, FOOD, positive, 24th May). It is clear that these hextuples remove any remaining sentiment ambiguity in comparison to their quintuple counterparts from the previous paragraph. Note, in general not all of the values of these hextuples must be filled but rather the more that are filled potentially less ambiguous the sentiment would be and more detailed analysis can be performed, e.g. sentiment over time or comparing sentiment of different holders on a particular target/aspect/entity.

Definition 1 *Hextuple fine grained sentiment definition: (h, t, a, e, s, ti). Where h = sentiment holder, t = target, a = aspect, e = entity, s = sentiment, and ti = time.*

Lastly, we state here some explanations and edge cases that come from the hextuple definition. The sentiment holder can be very important in sentiment disambiguation as shown by example 7 sentence 2, which without the sentiment holder information it would be unclear what the sentiment value is for 'talks', as Lucy found it negative and Joe positive. However, Liu (2015) motivated the reason for the sentiment holder[44] within the sentiment definition is due to the potential influence difference between different holders, e.g. an online influencer sentiment towards a target/product could be more important than an everyday customer. Within this thesis the importance of influence of the holder is recognised for applications. From a sentiment perspective example 7 sentence 2 shows the main reason for the holder being required in the sentiment definition, to resolve sentiment ambiguity. Additionally, the target within the hextuple can be empty due to the target being, implicit thus this is when the aspect-entity pair is required to resolve what the implicit target is, this is shown on sentence 3 in example 7. Also, the aspect within the aspect-entity pair is not always required if a hierarchical structure is not required as only the entity is needed to resolve implicit target situations, as shown in sentence 3 of example 7. Finally, as shown in example 7 sentence 1, the time element could be required to disambiguate the sentiment, however the time effect is a very small edge case.

In comparison to the latest SemEval fine grained sentiment analysis task, SemEval 2016 (Pontiki et al., 2016), their definition of this task is a quadruple of; (target, aspect, entity, sentiment). This quadruple, as motivated by the previous paragraph, does not contain enough information to remove all sentiment ambiguity because of not having the time and more importantly the sentiment holder. Hence, showing why the hextuple defined in the thesis is a more complete definition for fine grained sentiment analysis. This concludes the motivation and reason behind the new fine grained sentiment definition of the hextuple, which can be seen as a direct extension and backward compatible with the previous definitions.

[44]This was on page 18 of Liu (2015).

*(1) Ben used to love **lectures** but no longer due to the new lecturer. (2) Lucy did not enjoy the **talks** but Joe did. (3) Dave loved **University** but found it expensive.*

Example 7: Made up fine grained sentiment example. This example contains the following hextuples; (Ben, lectures, LECTURES, EDUCATION, positive, past), (Ben, lectures, LECTURES, EDUCATION, negative, present day), (Lucy, talks, LECTURES, EDUCATION, negative, past), (Joe, talks, LECTURES, EDUCATION, positive, past), (Dave, university, -, EDUCATION, positive, present), and (Dave, -, PRICE, UNIVERSITY, negative, present). The dash (-) symbol within the hextuple represents the value not existing.

Datasets that could make use of part of these hextuples are the popular SemEval 2015 and 2016 restaurant datasets, which contain (target, aspect, entity, sentiment) values of the hextuple. This thesis has found that without including the target within the hextuple it would be impossible to classify 3.68% and 3.27% of samples correctly from the 2015 and 2016 datasets respectively due to sentiment ambiguity[45]. This is due to texts containing the same aspect-entity pair more than once with different sentiments. Even though these percentages are small it further empirically justifies why it is needed to include the target within the hextuple definition.

Wiebe (1994) described that the private state, which can be seen as the original fine grained sentiment definition, can only be found in subjective sentences. Thus the fine grained sentiment cannot be used within objective sentences or in objective cases. In this thesis it is argued that objective sentences can contain fine grained sentiment as stated in section 2.4.2 of Liu (2015). Example 8 shows an objective negative sentiment towards Rio tinto, this is also an example of implicit sentiment which is discussed in more detail in subsection 2.5.1.

*(1) **Rio tinto** shares went down on the 22nd of May.*

Example 8: Made up fine grained objective sentiment example. This example contains the hextuple (Author, Rio tinto, MINING, STOCKS, negative, 22nd of May).

Given this hextuple definition 1 it is clear how sentence and document ABSA from the last two subsections 2.3.3.2 and 2.3.3.1 can be ambiguous with respect to sentiment. This is best shown in example 7 whereby in sentence 1 for the aspect-entity pair (LECTURES, EDUCATION) there is both a positive and negative sentiment, thus from a ABSA perspective it would be impossible to capture both sentiments. This case will become more problematic with longer texts e.g. document ABSA, as they are more likely to contain multiple of the same aspect with different sentiment. Hence as stated in the opening paragraph of sentence ABSA subsection 2.3.3.2, document ABSA in effect summarises the sentiment information of aspects, and this would have to happen when ambiguities occur at the sentence level as well as is the case within example 7 sentence 1. To make this more concrete, the aspect summary for the aspect-entity pair (LECTURES, EDUCATION) in sentence 1 of example 7 could be neutral as there is both a positive and negative sentiment towards the aspect-entity pair.

The main tasks to create the hextuple are the following:

[45]URL to the python notebook that reproduces this result `https://bit.ly/3gtx1RN`.

Task 1 Target extraction.

Task 2 Entity and Aspect mapping. Due to the cases of implicit targets, one would have to treat this task as ABSA extraction[46] for the given text, as one cannot assume that a target is always in the text. Then given the aspect-entities that have been extracted for the text map them to the targets extracted as well as a special implicit target symbol if implicit target(s) exists. This is only one suggestion on how to perform this task that takes into account implicit targets.

Task 3 Sentiment holder extraction.

Task 4 Link the relevant target, holder, and entity, aspect pairs within the given text together.

Task 5 Extract the time given the holder, target, and entity, aspect pair. Time extraction would more than likely come from the metadata e.g. the timestamp from a Tweet. However time extract could be complex if the text is a narrative/story, thus has to be extracted from within the text. Furthermore as in sentence 1 of example 7, time can be target and holder specific where in this case there is both a past and present day timestamp.

Task 6 Extract the sentiment associated to the target, holder, aspect, entity pair, and time. Once the sentiment has been resolved dependent on the other values in the hextuple, the hextuple has been created.

These tasks can be done either in a pipeline setup or where appropriate in a joint multi-task setup. The tasks are not ordered, but the tasks do have dependencies, for example task 4 cannot be done without already completing the first three tasks. These tasks are similar to those in Liu (2015)[47] but not completely the same. Liu (2015) does not contain task 1 as target extraction is not within their quintuple definition. Tasks 2, 3, 5, and 6 map respectively to Liu (2015) tasks 1-4. Task 6, sentiment extraction task, is different to Liu (2015) task 5, as they only take into account the aspect, entity pair where as in this task all other values of the hextuple that have been found and linked together from the preceding tasks is taken into account.

This thesis's main focus is target dependent sentiment analysis (TDSA), which so far has not been defined. Even though the definition of fine grained sentiment analysis has been well stated, this was required to fully understand where TDSA fits into sentiment analysis and assumptions that are made in the task. TDSA's objective is a simplification of task 6, instead of extracting the sentiment with respect to the target, holder, aspect, entity pair, and time the only element of the hextuple that is taken into account is the target.

In the rest of this section the tasks described above will be reviewed. Task 2 of entity and aspect mapping will not be reviewed as it is believed that none have performed entity and aspect mapping. However there are two very related strands of research, that of ABSA extraction (Pontiki et al., 2016), and implicit target extraction which will not be reviewed here but will point the interested reader to this survey paper (Ganganwar

[46] This task is described as aspect category detection in Pontiki et al. (2016).
[47] See pages 26-27 of Liu (2015).

et al., 2019). Task 4 will also not be reviewed as again it is believed that none have performed this task fully due to the lack of work in task 2. However within the holder extraction subsection, work will be discussed that has been performed that performs both holder (task 3) and target (task 1) extraction. Lastly task 5 will also not be reviewed, as again time is not normally a problem that is resolved within sentiment analysis, nor is it believed to be a problem that would affect the sentiment ambiguity of a target. Furthermore, in a lot of use cases the time element could be extracted from the metadata as stated before or if this is not given from the text itself. However extracting time from the text in itself is a research area whereby the time element does not have to be just a date or time (Bethard et al., 2016; Viani et al., 2018). Thus as the time element is covered by either metadata or a different but somewhat related area this work will not be covered in this thesis. Task 6 will not be reviewed as it requires all previous tasks to be completed, of which not all prior tasks have been explored, e.g. tasks 2, 4, and 5.

2.4.1 Target Extraction

Target extraction is the task of extracting objects that are contained within the text, where these targets have some sentiment associated towards them (Hu et al., 2004b; Wilson, 2008, Chapter 7). This definition is fairly consistent across the whole field. There is some inconsistency between datasets on the finer details of what is defined as a target, for instance the SemEval datasets (Pontiki et al., 2014; Pontiki et al., 2015; Pontiki et al., 2016) explicitly do not consider pronouns to be targets, whereas others do (Toprak et al., 2010; Kessler et al., 2010). These subtle differences could be important depending on the use case, thus reading the annotation guidelines associated with these datasets is always advisable. Target extraction has also been called different names by different prior works over the years: feature extraction (Hu et al., 2004b), opinion target extraction (Qiu et al., 2011), and more recently aspect term extraction (Pontiki et al., 2014).

One of the first works in this area was by Hu et al. (2004b) where they state four main points on why a list of prior known targets is not feasible, and their four points can be summarised as the following two points: i) not all targets are known ahead of time, and ii) targets can be known by different names e.g. laptop as notebook, and targets can be written differently e.g. misspelling or abbreviations etc. Given this, a list of prior targets cannot be a feasible solution to the target extraction problem.

Many of the earlier works in this area use unsupervised rule based systems. Hu et al. (2004b) used POS and chunking information to find frequently occurring noun and noun-phrases that are then pruned based on frequency rules, which are then considered frequent targets. Infrequent targets are found by labelling the nearest noun or noun-phrases that occur closest to an adjective that is within their in-domain sentiment lexicon. This in-domain sentiment lexicon is created by extracting the nearest adjective to all of the frequent targets. This was improved upon by (Popescu et al., 2005) by incorporating Pointwise Mutual Information (PMI) (Church et al., 1989) between the frequently occurring noun phrases and the known product name being reviewed through a large information retrieval system. They show that using more external data, rather than just the task data, within the information retrieval system, greatly improves results.

Qiu et al. (2011), similar to Hu et al. (2004b), created a syntactic approach whereby targets are found based on the relationship with the sentiment words that affect the target. By using just a few seed sentiment words they iteratively extract targets through

dependency relation rules which in turn they use to find more sentiment words, this process is repeated until no more sentiment or target words are found. Liu et al. (2012) used a word based translation model to find the relationship between sentiment words and targets avoiding the need for a dependency parser and the error it can introduce into the modelling. Using this approach, unlike the previous one, they can find many-to-one and one-to-many relationships between sentiment and target words. This approach was shown to outperform the existing unsupervised techniques on the Hu et al. (2004b) customer review dataset, which at the time was the standard benchmark dataset.

These unsupervised rule based approaches, which are believed to have no code releases, are questionable in terms of how easy they are to reproduce (Marrese-Taylor et al., 2017b). Marrese-Taylor et al. (2017b) found that they could not reproduce the results from Hu et al. (2004b), Qiu et al. (2011), or Liu et al. (2012) with the best reproduced result being circa 50% of the original. In general, they found that key parameter settings were not stated in the papers, which lead to larger parameter searches. This therefore brings into question how good these rule based approaches actually are if it is not possible to reproduce them.

The most popular and currently best approach to target extraction are supervised sequence labelling methods. Sequence labelling methods generally predict the current token label based on its context, whereby the context can be the entire text and all previous token label predictions. Due to the nature of the task being one of extracting potentially multi-word targets the tag set used in sequence labelling has to be one that can allow multiple words to be extracted, thus the BIO tagging scheme is commonly used (Liu et al., 2015). Example 9 demonstrates the BIO tagging scheme[48].

The$_O$ ***chicken$_B$*** ***pot$_I$*** ***pie$_I$*** *is$_O$ excpetiona,$_O$ the$_O$* ***cheeseburger$_B$*** *huge$_O$ and$_O$ delictable,$_O$ and$_O$ the$_O$* ***service$_B$*** *professional$_O$ wan$_O$ warm.$_O$*

Example 9: Target extraction example demonstrating the BIO tagging scheme, whereby all the target words are in **bold**. This was taken from the SemEval 2015 restaurant dataset (Pontiki et al., 2015), sentence id 1264954:2.

One of the first sequence labelling approaches was that of Jin et al. (2009), where they used a Hidden Markov Model (HMM) which had a context window of the current word and the previous, using POS tag and lexical[49] information as well as the previous word label to predict the current label. They found their approach, compared to unsupervised methods, generalised better to new and infrequent/rare targets, as well as finding targets that are not just noun or noun-phrases, which the prior approaches assumed targets must be. These improvements compared to the unsupervised rule based approaches most likely are gained from the model allowing to learn internal linguistic rules within the black box without having large assumptions imposed on it like a target has to be noun or noun-phrase.

The field progressed from using HMM to using Conditional Random Field (CRF) (Lafferty et al., 2001), and the two best performing target extraction models at SemEval 2014 (Pontiki et al., 2014) were CRF based approaches using lexical, syntactic, and semantic features (Chernyshevich, 2014; Toh et al., 2014). The drawback to these

[48]BIO can also be referred to as the IOB2 tagging scheme.
[49]The word itself.

supervised approaches is that they all require additional linguistic features e.g. POS tags of which finding which features are useful (feature engineering) is time consuming. Further, some of these features might not be available in all languages. Liu et al. (2015) showed that an RNN/LSTM, a form of NN, combined with word embeddings can learn the features required to be competitive with the feature engineered CRF approaches. They further showed that when the LSTM is combined with additional linguistic features, POS tags and chunk information, the results for one of the datasets outperforms the CRF approaches. Since Liu et al. (2015) many approaches started to use RNNs/LSTMs and Recursive NN (RCNN) to learn feature representations.

Many approaches have combined extracting targets with extracting the sentiment words within the text as a joint or multi task learning (MTL) setup. In all of these cases they have found performing both tasks to improve results and in all works they have used either a GRU with coupled attention (Wang et al., 2017c), LSTM with attention (Li et al., 2018c), LSTM with memories (Li et al., 2017), GRU stacked on a CNN (Jebbara et al., 2016)[50] or RCNN (Wang et al., 2016a). The assumption the models make is that by jointly learning the two tasks it can better extract the targets, as targets have some form of sentiment associated with them, thus the sentiment words must modify the targets. This approach can be seen as a supervised approach of Qiu et al. (2011) and Hu et al. (2004b), whereby the model learns the relationships between sentiment words and targets. In most cases the joint learning approach is setup using the BIO tag set but the targets and sentiment words have a different category label e.g. t for target and s for sentiment words as shown in example 10. For clarification, in none of these case are they explicitly linking the sentiment words with any of the targets nor are they learning the sentiment label of the sentiment words. Li et al. (2017) also found predicting if the current sentence contains a target to be a useful auxiliary task within a MTL setup[51], similar to the opinion sentence feature within Jakob et al. (2010a) CRF.

The_{O} $\textbf{chicken}_{B\text{-}t}$ $\textbf{pot}_{I\text{-}t}$ $\textbf{pie}_{I\text{-}t}$ is_{O} $excpetiona,_{B\text{-}s}$ the_{O} $\textbf{cheeseburger}_{B\text{-}t}$ $huge_{B\text{-}s}$ $and_{I\text{-}s}$ $delictable,_{I\text{-}s}$ and_{O} the_{O} $\textbf{service}_{B\text{-}t}$ $professional_{B\text{-}s}$ $wan_{I\text{-}s}$ $warm._{I\text{-}s}$

Example 10: Target and sentiment word extraction example demonstrating the BIO tagging scheme with target and sentiment labels as t and s respectively. All the target words are in **bold** and the sentiment words are highlighted. This was taken from the SemEval 2015 restaurant dataset (Pontiki et al., 2015), sentence id 1264954:2., whereby the sentiment words were annotated by the author of the thesis.

It is clear from these prior works that performing target extraction with sentiment word prediction is useful. However two of these prior works (Wang et al., 2016a; Wang et al., 2017c) require human annotation of the sentiment words, which is costly, but these annotations have been made available by the authors[52]. Li et al. (2017) and Li et al. (2018c) used the MPQA sentiment lexicon (Wilson et al., 2005) instead of human annotators as a form of noisy sentiment word labels. It is interesting that models like

[50] Jebbara et al. (2016) state in the paper the performance increase might not be due to the joint target extraction and sentiment word extraction but rather the larger parameter size of that model.

[51] They describe this task as predicting a sentimental sentence.

[52] For Wang et al. (2016a) the annotations can be found `https://bit.ly/3cMbrok`. For Wang et al. (2017c) the annotations can be found here `https://bit.ly/2zh5CBE`.

LSTMs that do not rely on feature engineering can incorporate a noisy signal from a sentiment lexicon to improve the model, whereas a CRF that relies on features using sentiment words cannot (Jakob et al., 2010a). It would be of interest to see the difference in performance between human sentiment word annotations and sentiment lexicon noisy annotations, and further the effect of different sentiment lexicons across datasets of different domains.

The LSTM based methods have improved upon the feature engineered CRF approaches without requiring any additional sentiment word information or linguistic features since Wang et al. (2017c)[53] and Li et al. (2017) baseline bi-directional LSTM. Li et al. (2017) was also the first to show that bi-directionality within LSTM based methods improved results, where previously Liu et al. (2015) found this not to be the case. The reason for the difference could be due to Liu et al. (2015) treating each input word vector to be based on the concatenation of the current, previous, and next word's word embedding, whereas Li et al. (2017) just used the current word's word embedding. Most works since Li et al. (2017) that have used LSTMs have used bi-directional LSTMs (Li et al., 2018c).

Most of these approaches have been fairly complex, either through incorporating dependency information into an RCNN (Wang et al., 2016a), different forms of attention to better incorporate sentiment word information (Wang et al., 2017c; Li et al., 2018c), and sharing information between multiple LSTMs through memories (Li et al., 2017) to again better capture sentiment word information. However, Xu et al. (2018) showed that using both an in-domain embedding with a general embedding through simple concatenation, which is then used as input to a CNN, outperforms all of these prior methods. For clarification, most of these prior works used an in-domain embedding (Wang et al., 2016a; Wang et al., 2017c; Li et al., 2017) but the in-domain embeddings from prior work used Word2Vec (Mikolov et al., 2013a) whereas Xu et al. (2018) used FastText (Bojanowski et al., 2017) which can create embeddings for out of vocabulary (OOV) tokens. This work demonstrates the gains that can be produced from using better pre-trained word representations in comparison to complex architectures and or human annotated sentiment word information.

The latest supervised approaches follow this trend of using better word representations through fine tuning BERT (Devlin et al., 2019) a CWR model[54]. Xu et al. (2019) showed that fine tuning $\text{BERT}_{\text{base}}$ itself is no better than using a combination of general and in-domain word embeddings (Xu et al., 2018). However when $\text{BERT}_{\text{base}}$ was adapted to the domain the results surpassed those of Xu et al. (2018). This shows that CWR can make large gains through adapting to the domain, which was also shown in the non-CWR work (Liu et al., 2015; Jebbara et al., 2016), and more broadly Gururangan et al. (2020)[55] has shown adapting CWR to the relevant domain to be effective across many NLP tasks and domains. Hu et al. (2019b) in comparison found that fine tuning $\text{BERT}_{\text{large}}$ outperformed both Xu et al. (2018) and in half the cases the domain adapted $\text{BERT}_{\text{base}}$ of Xu et al. (2019). These results are not fully comparable as Hu et al. (2019b) used span prediction rather than a sequence labelling model. Further, the biggest difference with respect to comparisons comes from the number of parameters in the model as $\text{BERT}_{\text{large}}$

[53] See the ablation results within table 4 that only uses a GRU with attention without sentiment word information.

[54] Fine tuning is explained in more detail within section 5.4.1.

[55] In Gururangan et al. (2020) pretrained language models is the equivalent to CWR.

contains $\approx 340M$[56] compared to $\approx 110M$ that $\text{BERT}_{\text{base}}$ contains. In general, results improve through having better word representations, but it is important to state the details of the CWR models used to allow for fair comparisons.

Jebbara et al. (2017) has created two quantitative error analysis splits for target extraction, the OOV split and multi word expression (MWE) split. The OOV split creates three subsets of the data *OOV op*, *OOV sent*, and *no OOV* based on a target word containing at least one OOV word, at least one word containing an OOV in the text, and no tokens in the text being OOV respectively. The MWE split creates subsets based on the number of tokens that make up the target, in Jebbara et al. (2017) they had three subsets each containing sentences that had targets that are at least two, three, or four tokens long respectively. They found in general *OOV op* and *OOV sent* to be equally difficult and harder than *no OOV*. Also discovered through the MWE split was that the longer (based on token length) the target the more difficult the extraction. Further, they found improvements on longer targets by adding character information into their word embedding based GRU method, which they believe is due to the character model being able to handle spelling variations. However the character information did not improve over what was expected on the OOV split, which they believe is due to the word embedding being trained on in-domain data.

The most common datasets used are the SemEval 2014 (Pontiki et al., 2014), 2015 (Pontiki et al., 2015), and 2016 (Pontiki et al., 2016) restaurant datasets, SemEval 2014 laptop dataset (Pontiki et al., 2014), and from earliest unsupervised work the customer review dataset (Hu et al., 2004b). A potential reason why the customer review dataset was not used as much within the latest supervised work could be due to annotation inconsistency (Marrese-Taylor et al., 2017b). Toprak et al. (2010) also noted this inconsistency, of which one can be seen in example 11, whereby the target stated in the annotation is battery but the character offsets are not given for the target, thus impossible to know without a human going through it which battery is the correct battery. The metrics commonly used to evaluate target extraction methods is the exact match F1, precision, and recall scores (Liu et al., 2015).

battery[+2]##the standard ***battery*** *include with the g3 is a camcorder* ***battery*** *that will allow me to take pictures all day without worrying about charging .*

Example 11: Example from customer review dataset, whereby battery[+2] indicates the target with positive sentiment is battery within the text after ##, no character offsets are given. All references to battery in the text are highlighted in **bold**. This was taken from Line 421 from Canon G3 text file.

2.4.2 Sentiment Holder Extraction

Sentiment holder extraction, from now on referred to as holder extraction, is the task of extracting the direct or indirect holders or sources of the sentiments expressed in the text (Choi et al., 2005). This task can be seen in example 12 whereby sentence 1 contains the direct sentiment holder Bella whereas sentence 2 contains an indirect sentiment holder as Christopher is spoken about through June. The task has also been called opinion source identification (Choi et al., 2005), source extraction (Choi et al., 2006), opinion holder

[56]M stands for million.

extraction (Johansson et al., 2010; Wiegand et al., 2012), and as part of the task of opinion entity identification[57] whereby the holder is one of the entities to identify (Yang et al., 2013).

*(1) Bella enjoyed her **visit to Llanfyllin**. (2) June said that Christopher did not like the **amount of traffic on the road to Llanfyllin**.*

Example 12: Made up example to show direct and indirect sentiment holders.

Choi et al. (2005) state that holders are i) typically noun phrases, ii) the holder phrase should be an entity that can semantically bear or express a sentiment, and iii) the holder phrase should be linked/related to a sentiment. From these three properties Choi et al. (2005) points out that the third property distinguishes holder extraction from the task of Named Entity Recognition (NER).

Choi et al. (2005) was one of the first to perform this task, whereby they framed it as a sequence labelling problem. They used a CRF with multiple syntactic and semantic features. Included in these features is if the current, previous, and next word is in a sentiment lexicon, and more notably whether the dependency head of the current word is part of a syntactic chunk that contains a sentiment word from the lexicon. These features cover the three properties that they state make up a holder.

Choi et al. (2006) showed that by combining Choi et al. (2005) CRF tagger for holder extraction, another CRF tagger for sentiment word extraction, and a feature based binary classifier greatly improves holder extraction compared to a non-combined system. Further, they state that sentiment words and holders have a one holder to a many sentiment word relationship. Johansson et al. (2010) also found that by modelling the relationship between sentiment words and holders improves holder extraction. These joint models were further added to by Yang et al. (2013) combining the extraction of holder, sentiment words, and targets using CRFs as well as the relations between the sentiment words and their respective targets and holders using a feature based logistic classifier. Both Johansson et al. (2010) and Yang et al. (2013) both took into account implicit sentiment word relationships whereby the sentiment word does not affect either a holder or a target, whereby Yang et al. (2013) shows large recall benefits within the relation task.

All of these current approaches require a mass of features and thus feature engineering. Katiyar et al. (2016) used a bi-directional LSTM to automatically create features while only using pre-trained word embeddings as input to the model. They found that the model could not outperform a feature engineered joint CRF model, but could exceed the non-joint version after the LSTM uses a CRF type of approach to learn dependencies within the output layer instead of the standard softmax. Marasović et al. (2018) believed that the lack of data was one of the reasons why the LSTM approach was not as successful as the feature engineered approach. They overcome this through MTL with Semantic Role Labelling (SRL) as the auxiliary task, this was chosen as many of the feature engineered approaches used SRL features (Choi et al., 2005; Johansson et al., 2010; Yang et al., 2013). They found MTL with SRL does improve performance over just the single task. This was further verified by Zhang et al. (2019b) which instead of creating an

[57] Opinion entity identification has also been called opinion entity extraction (Katiyar et al., 2016) and opinion role labelling (Marasović et al., 2018; Zhang et al., 2019b)

MTL model with SRL whereby they share all layers in the LSTM, added final output of the SRL model to the first layer of the holder/target extractor. They state that the improvement came as they do not share all layers with the SRL, thus there is less error propagation from the SRL.

The main dataset for holder extraction is the MPQA 2.0 corpus (Wiebe et al., 2005). There are also several evaluation metrics on a range of difficulties whereby the most strict is exact match F1, followed by proportional, and then the easiest binary[58] F1 (Zhang et al., 2019b; Marasović et al., 2018; Katiyar et al., 2016). Both Katiyar et al. (2016) and Marasović et al. (2018) found that their models outputted correct labels but the human annotations from the MPQA 2.0 corpus had not labelled these as correct suggesting that annotations might be missing from the MPQA corpus. Marasović et al. (2018) found that their models struggled with holders and targets that are far away from their related sentiment words, thus suggesting dependency parsing could be a useful auxiliary task to overcome these long-range dependency problems. As Marasović et al. (2018) main motivation for MTL was to overcome the lack of data, an alternative to MTL would be to use CWR[59] as suggested by Marasovic (2020) in section 5.3.

2.4.3 Target Dependent Sentiment Analysis

TDSA is the task of predicting the sentiment of a target with respect to the text it is within. Unlike ABSA a target is not latent and thus has to always exist within the text. Like other fields TDSA has also been called by other names, of which the most frequent of these is aspect based sentiment analysis (Wang et al., 2016b). TDSA is not the same task as targeted sentiment analysis, targeted sentiment analysis involves the joint task of extracting and predicting the sentiment of a target (Mitchell et al., 2013), TDSA only involves the latter task. One of the major differences between TDSA and sentence/document level sentiment analysis is target specific sentiment, whereby most targets are affected differently based on the sentiment words affecting it, even within the same domain. An example of this is: *The **knives** were sharp*. From this example *sharp* can easily affect a target positively or negatively, an example where it can affect a target negatively in the same domain: *The edge of the table were sharp*.. This target specific sentiment can also happen for aspects/entities when the target is implicit within fine grained sentiment analysis and for aspects in ABSA. This is stated in Ding et al. (2008, §2) and Popescu et al. (2005, §3.3.4).

Nasukawa et al. (2003) was one of the first to create a TDSA system in which they created a rule based method using a manually created sentiment lexicon, POS tags, chunking[60], and dependency parsing[61] information. This was then followed by (Hu et al., 2004a) which instead of using a manually created sentiment lexicon, they automatically create one using a few known seed adjectives[62]. They bootstrap off the seed adjectives to label other adjectives, which are synonyms or antonyms using WordNet (Miller, 1995) of these seed adjectives, with the seed's sentiment labels. The sentiment of each target is then labelled with the nearest adjective that has been assigned a sentiment label from

[58] This is sometimes called overlap instead of binary Choi et al., 2005; Choi et al., 2006
[59] In Marasovic (2020) thesis they use the words pre-trained language models instead of CWR.
[60] In the paper they use shallow parsing, of which this is interpreted as chunking.
[61] In the paper they use syntactic parsing, of which this is assumed to be dependency parsing.
[62] This lexicon can be found here `https://bit.ly/2Y3iJik`.

the expanded sentiment lexicon. Even though the main sentiment part of the paper was sentence sentiment analysis this technique of labelling targets was used to help identify infrequent targets in the paper and can be used as a TDSA method. Furthermore, one assumes this TDSA approach was the one used in Popescu et al. (2005) and Ding et al. (2008) when comparing methods.

Popescu et al. (2005) used a set of dependency rules to extract sentiment words associated with a target, those sentiment words were then labelled with an initial sentiment based on Turney (2002) unsupervised sentiment PMI (Church et al., 1989) approach. Lastly those sentiment words were contextualised with respect to the target and sentence using relaxation labelling (Hummel et al., 1983), an unsupervised technique, which finds the most likely sentiment based on a set of constraints. The constraints in this case come from the target, dependency rules, conjunctions, synonym and antonym, and morphological relationships. Ding et al. (2008) also created a rule based system, they extended the sentiment lexicon from Hu et al. (2004a) to include verbs and nouns using the same bootstrapping technique and also manually created an idiom sentiment lexicon. Further they automatically generated a target dependent sentiment lexicon based on linguistic rules. A target is then scored based on a distance weighted sentiment lexicon count taking into account negation and conjunctions through a set of rules. Sentiment for implicit targets was taken into account by assuming some adjective sentiment words imply a target. Ding et al. (2008) compared their system to the other unsupervised rules based systems (Hu et al., 2004a; Popescu et al., 2005) across Hu et al. (2004a) dataset and their own dataset which used the same annotation guideline as Hu et al. (2004a).

The approaches described so far have all been rule based and unsupervised, this approach similar to how the target extraction literature and more broadly most NLP topics was overtaken by supervised methods. These supervised approaches could also benefit from not requiring feature engineering through using different NN architectures learning the required features. RCNN became popular whereby these made use of the dependency tree structure (Dong et al., 2014; Nguyen et al., 2015). Where as some used Neural Pooling which averaged word embeddings based on their various different windows of text that relate to the position of the target in the text (Vo et al., 2015; Wang et al., 2017a). LSTMs have also been used taking into account the position of the target (Tang et al., 2016b). LSTMs have also been combined with attention to better model longer term dependencies (Wang et al., 2016b; Chen et al., 2017).

Even though TDSA has been well explored through various different methods. None have attempted a reproduction study to evaluate if a paper can be recreated only from what is written within the paper. This may seem unnecessary due to the field moving towards open sourcing code with papers. However, even when the code is open sourced it would appear that results can differ, as shown by Chen et al. (2017) generating better results than the original (Tang et al., 2016b). Furthermore, Tay et al. (2018) found that when they re-implemented the paper they got a lower result than original (Tang et al., 2016b). Thus motivating the need for a reproduction study.

Further, many of these papers are evaluated on a limited number of datasets (as shown in chapter 3). Thus when the authors may state, state of the art performance it is not known to what extent they are state of the art. Thus a large scale exploration into what is state of the art across many datasets and different methods is required to allow the field to better know the limitations of different methods.

Lastly, very few of these works perform detailed quantitative error analysis, most only report the metric results applied to the whole dataset. Further from the existing quantitative error analysis techniques/splits (Nguyen et al., 2015; Wang et al., 2017a; Zhang et al., 2019a) that do exist no detailed work has been performed to analyse what they actually show and when they are useful.[63]

The main datasets for TDSA are the SemEval 2014 (Pontiki et al., 2014), 2015 (Pontiki et al., 2015), and 2016 (Pontiki et al., 2016) laptop and restaurant datasets[64]. Another popular dataset is the Twitter dataset of Dong et al. (2014). The most popular datasets during the unsupervised methods are the Hu et al. (2004a) and Ding et al. (2008) customer review datasets. Additional datasets for TDSA include the MPQA 2.0 corpus (Wiebe et al., 2005) and the review corpus by Toprak et al. (2010). For a more detailed analysis of a large range of TDSA datasets see section 3.4.1. The main evaluation metrics are accuracy and macro F1, whereby macro F1 better takes into account any unbalanced label distributions in the evaluative dataset.

2.5 Further Related Topics

These topics are not extended directly within the thesis, but are closely related to the overall topic of fine grained sentiment analysis. The literature review of the following topics will be brief but will contain references to papers that will allow the interested reader to explore the topic in more detail. The main reason for including these topics is to give the reader a primer into parts of the future work section 6.3.

2.5.1 Implicit and Factual Sentiment

In this thesis it has not been stated whether the sentiment that is being evaluated is explicit or implicit, this additional information has been largely ignored so far in previous research due to most datasets not stating which kind of sentiment it is via the annotation. Explicit and implicit sentiment usually come from subjective and objective texts respectively (Toprak et al., 2010; Kauter et al., 2015), however in the MPQA 2.0 corpus (Wiebe et al., 2005) expressive subjective elements are still defined as subjective even though their sentiment is not explicit. Thus implicit sentiment cannot be defined as objective or subjective sentiment. Implicit sentiment has been defined by (Russo et al., 2015) as "the recognition of subjective textual units where no polarity markers are present but still people are able to state whether the text portion under analysis expresses a positive or negative sentiment"[65]. Explicit, on the other hand is the opposite where the sentiment of the text is clear based on the sentiment bearing words that appear in the text. Example 13 and 14 show cases of implicit and explicit sentiment respectively.

I know that I have my disease under control .

Example 13: Example of implicit positive sentiment towards the event/aspect **(FEAR_OF)_PHYSICAL_PAIN**, taken from the CLIPEval corpus (Russo et al., 2015).

[63] A more detailed literature review of error analysis for TDSA can be found in section 4.2.1.
[64] Only 2014 for the laptop dataset.
[65] Polarity here can be interchanged with the word sentiment.

I have no complaints about the entire PhD journey and highly recommend this school.

Example 14: Example of explicit positive sentiment towards the both the **PhD journey** and **this school** targets, taken from the Toprak et al. (2010) corpus.

Factual sentiment in this thesis is the same as implicit, it has been defined by Toprak et al. (2010) as "facts which can be objectively verified, but still imply an evaluation of the quality or value of an entity or a proposition", whereby an entity here is considered equivalent to a target in TDSA. From that definition it is clear that factual sentiment occurs implicitly as the sentiment/value is implied by the fact. Example 15 shows that factual sentiment is the same as implicit sentiment.

*E*trade requires no such fee*

Example 15: A factual positive sentiment towards the company **E*trade**, which has come from the Toprak et al. (2010) corpus.

Implicit sentiment occurs more often in newswire datasets such as the financial dataset of Kauter et al. (2015), where as explicit sentiment appears more often in review datasets (Toprak et al., 2010; Mæhlum et al., 2019; Øvrelid et al., 2020). The differences in the domains is most likely the reason for these differences as news content contains more objective and factual content, where as review datasets are a lot more subjective (Kauter et al., 2015). Furthermore in the financial literature, sentiment tends to be referred to as tone (El-Haj et al., 2019), where the tone tends to be more objective than subjective as can be seen in example 16. Thus could be seen as implicit sentiment within the categorisation in this thesis.

our profit margins increased despite higher raw material prices

Example 16: Positive sentiment/tone sentence, taken from El-Haj et al. (2016) financial corpus.

The CLIPEval (Russo et al., 2015) and the human need (Ding et al., 2018) corpora both contain only English implicit sentence level ABSA sentiments[66]. MPQA 2.0 (Wiebe et al., 2005) contains implicit sentiment expression denoted as expressive subjective elements in English. Deng et al. (2013) created a good for/bad for English dataset where by the implicit sentiment of a target is inferred from another object's perspective rather than the speaker or writer as is the case of TDSA, this can be best seen in example 17. The MPQA 3.0 English dataset (Deng et al., 2015b) contains both explicit and implicit sentiment towards eTargets[67]. TDSA datasets have also been annotated for implicit and explicit sentiment in English (Toprak et al., 2010; Kauter et al., 2015) and Dutch (Kauter et al., 2015)[68]. The Norwegian sentence level dataset (Mæhlum et al., 2019) that has been labelled for evaluative and non evaluative, whereby evaluative indicates that a sentiment exists in the sentence, has been annotated such that evaluative sentences have

[66]The human need corpus uses out of context phrases rather than sentences.

[67]They are called eTargets rather than targets, as an eTarget is only one token of a whole target span in which it is the head of the noun/verb phrase within that span. This is a simplified explanation, more details can be found within Deng et al. (2015b).

[68]The Kauter et al. (2015) dataset does not appear to be public.

been separated based on the implicit sentiment being personal or not. The reason for the differentialisation between personal and non-personal implied sentiment, whereby the explicit sentiment is also with the personal implicit sentiment, is so that the explicit and personal implicit is better separated from the implicit non-personal sentiment[69]. This Norwegian dataset was further annotated with TDSA annotations (Øvrelid et al., 2020). From the financial domain, El-Haj et al. (2016) created a dataset where the tone of sentences within UK preliminary earning announcements have been annotated[70].

Luckily Bill didn't kill him.

Example 17: Good for the target **him** where by the object called an agent, **Bill**, did not kill him and thus is beneficial for **him**. This was taken from the Deng et al. (2013) corpus.

An early method used a lexicon based approach with linguistic rules to better identify implicit sentiment that is created from nouns and noun phrases (Zhang et al., 2011). In contrast, both Deng et al. (2014a) and Deng et al. (2014b) used a rule based approach to classify implicit targets, and Deng et al. (2015a) also used a rule based approach where they first identified the holder and target then predicted the sentiment of target with respect to the holder. Mæhlum et al. (2019) used a NN approach and found that identifying non-personal implicit sentiment sentences much more difficult than explicit and personal implicit sentiment sentences, however the non-personal implicit sentiment sentences were a very small minority class (3.77%). Finally, both İrsoy et al. (2014) and Han et al. (2019) found that implicit sentiment expression identification is a more difficult task than that of explicit sentiment extraction. It is clear from the literature that understanding exactly the data you have is important, as techniques that work on review datasets that make use of explicit subjective knowledge are unlikely to work on more objective datasets, such as those in the financial domain and vice versa. Finally Xiang et al. (2019) has currently achieved SOTA on the CLIPEval dataset (Russo et al., 2015) whereby they use a GRU multi headed attention based NN similar to Wang et al. (2016b) ABSA NN.

2.5.2 Discourse level considerations within Fine Grained Sentiment Analysis

Within the literature review so far most of the methods for fine grained sentiment analysis (2.4) stated do not take into account anything beyond the sentence level information. One reason for this could be that some of the popular datasets do not require inter-sentence knowledge for predicting the relevant properties (Pontiki et al., 2014; Øvrelid et al., 2020). Furthermore, some of the datasets that do state that inter sentence information is necessary do not annotate which annotations require this information (Pontiki et al., 2015; Pontiki et al., 2016). The work reported below explores fine grained sentiment analysis beyond the sentence level utilising discourse information (Webber et al., 2003).

Taking discourse information into account would allow methods to potentially better contextualise words with respect to the whole text. For instance Kessler et al. (2009)

[69]The implicit non-personal sentiment in the paper were called *FACT-NP*.

[70]This dataset also contains tone towards two attributions, internal and external to an organisation. This can be viewed as sentiment towards two very high level aspects within sentence ABSA.

found that 14% of targets are pronouns, Jakob et al. (2010b) further showed on another corpus (Zhuang et al., 2006) that 504 targets[71] are pronouns but the corpus contained over 11,000 pronouns. This shows to some degree why additional information is required, such as co-reference information as knowing what the referent of the pronoun is can help disambiguate whether the pronoun is a target or not, as can be seen in example 18. Additionally this example shows that a pronoun does not have to be in the same sentence as the referent. Jakob et al. (2010b) showed that using a co-reference system can indeed improve the performance of a rule based target and sentiment expression extraction system. Thus showing the importance of taking additional discourse information into account. Another case where co-reference could be of use is resolving the holders of the sentiment, as shown in the error analysis of Marasović et al. (2018) in table 9 row 7.

yesterday i helped me mom with brians house and then we went and looked at a ***kia spectra****.* ***it*** *looked nice, but when we got up to* ***it****, i wasn't impressed. it looked like* ***it*** *was in an accident.*

Example 18: Multiple sentences taken from review car-001-027 from batch 1 of the JDPA car corpus (Kessler et al., 2010). All the bold pronouns represent the same target, which is the kia spectra.

Additionally, co-reference resolution can be of use to help group targets together to allow an end user to get a better understanding of what a target (and all of its references in the text) is positive or negative towards (Marasovic, 2020)[72]. In effect, this is a more fine grained version of the aspect of a target, where the main difference is that an aspect may contain targets that do not refer to the same referent. Stoyanov et al. (2008)[73] found that by creating a noun-phrase (NP) co-reference resolution system they could identify which targets belong to the same aspect, but their system did not perform aspect labelling just grouping of targets.

Lastly, discourse information can be of use to disambiguate the sentiment. Kessler et al. (2010) found 9% of sentiment expressions are linked to a target that is in another sentence. Thus without looking further than the current sentence the limit on any method would be 91%. This phenomenon can be seen in example 19 whereby the sentiment of the target ('Mac') is in a different sentence[74]. Somasundaran et al. (2008b) and Somasundaran et al. (2009b) also motivated that discourse information is required for disambiguating some targets' sentiments due to their reliance on other targets' sentiments in the same discourse[75]. They created an "opinion frame" corpus (Somasundaran et al., 2008a) which consists of triplets (sentiment 1, sentiment 2, same or alternative)[76] which denote if two targets represent the same or alternative target, where the condition for same is if the two targets refers to the same entity[77]. These opinion frames can then be

[71] 10% of all targets in the corpus.

[72] See section 1.3 of the thesis (Marasovic, 2020).

[73] In the paper they use the term 'topic' which is the same as aspect within this thesis. Further they define topic and target spans, whereas in this thesis both would be defined as the target span.

[74] This example was taken from Pontiki et al. (2015) paper.

[75] Very similar to the concept of implicit sentiment from subsection 2.5.1.

[76] Whereby sentiment 1 and 2 refer to the sentiment of target 1 and 2 respectively. The value for sentiment 1 and 2 could be positive or negative.

[77] Entity here is the same as an entity in co-reference resolution.

used to know if two targets are reinforcing each other or not, where reinforcing means that the sentiment of both targets are not contradicting each other just like a pros and cons list creates contradicting sentiment to the relevant target. They found that by adding features that incorporated reinforcing/non-reinforcing and alternative/same relations into a classifier for predicting target sentiment greatly improved results, especially for targets that have these relations (Somasundaran et al., 2009c). Thus showing adding discourse level relations between targets is important for fine grained sentiment analysis. These relations between targets is similar to inter-target encoding (Hazarika et al., 2018), but the key difference is that the inter-target encoding was only done within sentences, whereas these relations took into account targets within and beyond the sentence. Further the notion of reinforcing is similar to that of comparative sentiment analysis (Varathan et al., 2017).

*I was so happy with my new **Mac**. for 2 months. Then the hard drive failed;*

Example 19: Negative sentiment towards the **Mac** due to the hard drive failing. This came from sentence ids 128:3, 128:4 and 128:5 of the laptop training dataset of Pontiki et al. (2015).

The use of co-reference with respect to better capturing targets and their sentiment has also been stated within Sukthanker et al. (2020)'s review of anaphora and co-reference resolution[78]. They also state that co-reference can be of use to help disambiguate targets across multiple texts/documents/reviews.

It is clear that very few datasets contain discourse information and when they do some of these datasets are not publicly available (Zhuang et al., 2006; Stoyanov et al., 2008; Kauter et al., 2015). Considering those that are publicly available, Toprak et al. (2010) dataset includes co-reference of targets and sentiment holders. However their inter-annotator agreement levels for the co-reference annotations was low[79] compared to another co-reference dataset (Passonneau, 2004). They believe this low inter-annotator agreement was due to the guidelines not specifying which instance of the entity (target or holder) in the text should be the referent. Thus the annotators may agree on which entity is the referent but the annotations could be to two different instances of the same entity. Kessler et al. (2010) also contains co-reference annotations. Somasundaran et al. (2008a) created a corpus of "opinion frames" containing discourse level relations between targets as stated earlier. Finally Ding et al. (2010) created a co-reference corpus within the review domain[80], where they showed that sentiment features can help with co-reference resolution on their new corpus.

2.5.3 Stance Detection

Stance "refers to an overall position held by a person toward an object, idea or proposition" (Somasundaran et al., 2010). Stance detection is normally set up such that given an object/topic predict if the text is for or against that topic, where the text normally

[78]Specifically section 9 of the review.

[79]Toprak et al. (2010) inter-annotator aggrement levels for co-reference annotations was a Krippendorf's α (Krippendorff, 2018) value of 0.29 compared to Passonneau (2004) who obtained α values between 0.46 and 0.74.

[80]The corpus does not appear to be public.

represents the position of the person who wrote it. Example 20 shows a 'for' position by the Twitter user who wrote the Tweet on the topic of legalisation of abortion.

The pregnant are more than walking incubators, and have rights!

Example 20: Tweet which is **for** the topic **legalization of abortion**. This example comes from the first example in Mohammad et al. (2016).

The reason for the relation to sentiment is due to the task setup, in that the model has to predict label based on the text and a topic whereby the topic is latent just like an aspect within ABSA. Second, sentiment can be used to predict stance (Somasundaran et al., 2009a; Somasundaran et al., 2010; Mohammad et al., 2017) and has been shown useful as an auxiliary task in a multi task setup (Li et al., 2019). Further it has also been shown within Augenstein et al. (2018) novel label embedded multi task work that the label embeddings between stance detection and sentiment tasks such as ABSA are close within the embedding space[81]. Also they found that one of the best auxiliary tasks for stance was TDSA. However, they did not show how useful stance detection was for ABSA or TDSA as they found a more useful combination of other auxiliary tasks which included other ABSA and TDSA data, fake news detection (Riedel et al., 2017), and multi genre natural language inference (Nangia et al., 2017).

One stance dataset that is of particular interest is that of Mohammad et al. (2017). They created a Twitter stance dataset which also included annotations for sentiment. The sentiment labels state whether the sentiment is aimed at the stance topic or another arbitrary topic, and an example of each can be seen in 21 and 22. A version of this dataset was also used in the SemEval 2016 task 6 stance detection competition Mohammad et al. (2016). For a more detailed overview of stance detection see the survey by Küçük et al. (2020), which also states the relationship between ABSA and TDSA with stance detection in section 2.1 of their survey.

Hillary is our best choice if we truly want to continue being a progressive nation. #Ohio #SemST

Example 21: Tweet which is **for** the topic **Hillary Clinton** and has a positive sentiment towards the topic. This example comes from the trial dataset of Mohammad et al. (2017).

U know what isn't funny? Male politicians deciding what women should do with their body's. #SemST

Example 22: Tweet which is **for** the topic **legalization of abortion** and has a negative sentiment towards another topic (this other topic is only labelled as other in the dataset). This example comes from the trial dataset of Mohammad et al. (2017).

Stance has also been studied within linguistics, where the definition of stance is similar to that of Somasundaran et al. (2010) stated above; "Stance is a public act by a social actor, achieved dialogically through overt communicative means, of simultaneously evaluating objects, position subjects (self and others), and aligning with other subjects,

[81] See figure 2 in Augenstein et al. (2018).

with respect to any salient dimension of the sociocultural field." (Du Bois, 2007). The main difference between the two definitions is that of Du Bois (2007) taking into account other people's stance. However it could be argued that everyone takes a stance and thus you will be implicitly aligned with other people with regards to the stance that the person has taken.

Even though the examples given here are for stance in relation to short texts, other domains may require the stance of the author towards a topic given an entire review/post[82]. In these cases it is clearer to see that stance is a much higher level task than fine grained sentiment analysis and can easily be seen as a document level ABSA task. This shows that stance on many levels is very similar to sentiment, if anything perhaps a more simplified task as it is usually setup as a binary for or against a single topic rather multiple topics which is typical in ABSA. Further it can be seen from previous work how TDSA is useful for stance prediction (Somasundaran et al., 2009a; Somasundaran et al., 2010). Fine grained sentiment analysis that can link the targets' sentiment to a holder can be more useful for stance in cases where texts can contain many different holders views on a topic. Potential future work could explore how fine grained sentiment analysis could uncover stance in texts that contain multiple stances for multiple holders for a single topic, such as political debates from Hansard. For a great overview of sentiment and stance see Somasundaran (2010) thesis especially chapters 7 and 8.

2.6 Conclusion

The literature review has covered multiple levels of sentiment analysis from coarse grained document/sentence level to fine grained sentiment analysis. The review has described each granularity in detail with a large focus on methods, and it was found that the majority of these were evaluated on English datasets making this review biased towards the English language. Each granularity of sentiment is shown to help explain the next level and the reason why an extra level of granularity is required. Within fine grained sentiment analysis a new definition was created that extended the current (Liu, 2015), whereby unlike the previous definitions the new hextuple removes sentiment ambiguity. Further, this sentiment ambiguity was shown empirically, rather than just theoretically, to affect the current definition of Liu (2015) through the popular SemEval 2015 (Pontiki et al., 2015) and 2016 (Pontiki et al., 2016) restaurant dataset.

Within the TDSA review, the main area of research of this thesis, it is clear that there has been a lack of rigour within the evaluation of the methods. As stated, very few methods report any quantitative error analysis, whereby the error analysis breaks the results up into meaningful error splits for TDSA. Furthermore, even within the English language a large focus of work has been concentrated on just a few datasets making it difficult to know which method is best, or the more likely case, when to use which method. Additionally it highlights how one particular popular NN TDSA method (Tang et al., 2016b) has been reproduced with varying degrees of success.

This review motivates the need to study existing methods across more diverse existing datasets to better understand if one method is best, or if different methods work better under different conditions. Additionally, a detailed review of existing quantitative error

[82] In Somasundaran et al. (2010) and Somasundaran et al. (2009a) they perform experiments on political and product posts respectively where the posts in general are more than one sentence.

analysis techniques to unpick what they evaluate should be carried out, and if any more techniques are required to better understand what errors still occur within TDSA. Lastly a replication study into existing TDSA methods will be of use to explain why different implementations of the same method can obtain varying results.

Chapter 3

Reproducibility and Generalisability of TDSA Methods

3.1 Introduction[1][2]

Within this chapter the terms reproduce and replicate will be used frequently. Thus replicate will be defined as "running the exact same system under the same conditions in order to get the exact same results as output" (Fokkens et al., 2013), essentially running the published code from the paper. Reproducing the results requires re-creating a system through different means which should "lead to the same overall conclusions rather than producing the exact same numbers" (Fokkens et al., 2013).

As highlighted within the literature review (chapter 2) there has not been a reproducibility study within the TDSA literature. To many this might not be a problem as most works publish their code alongside the paper as shown by table 3.1. However, many of these codebases lack detailed documentation and mainly only provide a script to re-run the experiment, which is perfectly fine, but does mean many of the details of the method are hidden within the codebase. These finer details could well be reported in the paper which overcomes this issue.

In this chapter, three papers are reproduced: two Neural Pooling (NP) methods[3] (Vo et al., 2015; Wang et al., 2017a) (which are not based on Neural Networks (NN)), and one LSTM based (Tang et al., 2016b). All three have published their code. It is shown for the two NP methods that scaling features, a setting which was not reported in the papers, but for one (Wang et al., 2017a) did mention in part in the documentation of the codebase[4], can cause significant differences in the results. Further, it is also shown

[1] All code that creates the evidence for this chapter can be found in this codebase: `https://github.com/apmoore1/thesis-chapter-5-linear-models`. The evidence for table 3.1 can be found in this codebase: `https://github.com/apmoore1/tdsa-paper-details`.

[2] A lot of this chapter is based on, extends, and in some places contains complete or paraphrased extracts from Moore et al. (2018).

[3] A Neural Pooling method is a method that aggregates vector values dimension wise. An example use case for NLP is where all N words in a text are represented as dense word vectors of dimension size m, applying a Neural Pooling method aggregating using the maximum value would return a single word vector of size m with the maximum value for each dimension across the N words. See Vo et al. (2015) for more information on Neural Pooling.

[4] The README which can be found here: `https://github.com/bluemonk482/tdparse`.

for the NP methods that the C-value within the SVM classifier used in the NP methods can also cause significant differences. This parameter is reported in Vo et al. (2015) but not Wang et al. (2017a). Additionally, for the LSTM based method it is shown that reported results are difficult to reproduce without taking into account random seeds, a factor that is never mentioned in the paper and has only recently been shown to be an issue with Neural Network (NN) based methods within NLP (Reimers et al., 2017). The distribution of results generated from different random seeds suggests a possible reason why some prior works, which have attempted to reproduce (Tay et al., 2018) and replicate (Chen et al., 2017) this LSTM method, reported different results to each other and the original authors. Lastly, a new investigation into using larger more general word embeddings finds that they are at least as good as the original smaller more task-oriented embeddings, resulting in a trade off between performance, efficiency, and convenience. These reproduction findings contribute to answering RQ 1 'what lessons can be learned from reproducing a method within TDSA?'.

Number of papers	Code published (%)	Code link in paper (%)	Code link not in paper (%)
31	16 (51.61%)	13 (41.94%)	3 (9.68%)

Table 3.1: Out of the 31 papers published between 2013 and 2019 from the relevant NLP conferences (ACL, EACL, NAACL, EMNLP, CONLL, COLING, AAAI, TACL, and IJCAI) and WASSA workshops, this shows how many of them publish their code. Out of those that release their code the number that put a link to their code in the paper compared to those that do not and have to be found through an internet search. Note the author's paper (Moore et al., 2018) is not included in the statistics.

The second half of the chapter explores RQ 2 'how generalisable are existing methods within TDSA?'. This research question was motivated in part by the fact that many methods do not make use of the datasets that exist[5] as shown by table 3.2. Additionally these existing datasets often vary by type (e.g. review, social media, or news), domain (e.g. products), medium (e.g. written or spoken), dataset size, language, and many other factors. Thus, evaluating a method only on a subset of existing datasets limits the conclusions that can be drawn from those evaluated methods as they might perform well on those evaluated datasets, but it is unknown if they perform well on datasets from a different medium, domain, etc. Therefore to answer the research question, the reproduced TDSA methods from the first half of the chapter, which were selected due to their methodological differences, are applied to six English TDSA datasets. These six datasets vary by the following classes: domain, type, and medium as shown by table 3.3 (see the experimental setup section 3.4 for more details on the datasets used). This is the first large scale TDSA experiment that has evaluated a range of methods across all three classes. Doing so allows the methods to be tested for generalisation as none of them were originally developed for all six datasets. This research finds that methods are more affected by dataset size and sentiment class distributions than type, domain, and medium. Further, when controlling the dataset size the LSTM methods are greatly affected in comparison to the NP methods. These findings have important consequences as they bring to light to some extent when to use which method.

[5]Or existed at the time.

	Datasets						
Methods	**1**	2	**3**	**4**	5	**6**	**7**
Mitchell et al. (2013)			✓				
Kiritchenko et al. (2014)				✓			
Dong et al. (2014)	✓						
Vo et al. (2015)	✓						
Zhang et al. (2015a)			✓				
Zhang et al. (2016)	✓	✓	✓				
Tang et al. (2016b)	✓						
Tang et al. (2016a)				✓			
Wang et al. (2016b)				✓			
Chen et al. (2017)	✓			✓	✓		
Liu et al. (2017)	✓	✓	✓				
Wang et al. (2017a)	✓					✓	
Marrese-Taylor et al. (2017a)				✓			✓
1=Dong et al. (2014), 2=Wilson (2008), **3**=Mitchell et al. (2013), **4**=Pontiki et al. (2014), 5=Chen et al. (2017), **6**=Wang et al. (2017a), **7**=Marrese-Taylor et al. (2017a)							
Social Media Reviews News Not Applicable							

Table 3.2: A tick denotes that the method in the row has been applied to the dataset in the column, from the method's original paper and not a replication/reproduction of the method. Methods in **bold** are those that are being reproduced in this chapter. The dataset numbers in **bold** are the datasets that the reproduced methods will be evaluated on in this chapter. The colours represent the type of the dataset, apart from not applicable which indicates the dataset did not exist when the method was created.

3.2 Related Work

Reproducibility and replicability have long been key elements of the scientific method, but have been gaining renewed prominence recently across a number of disciplines with attention being given to a 'reproducibility crisis'. For example, in pharmaceutical research, as little as 20-25% of papers were found to be replicable (Prinz et al., 2011). The problem has also been recognised in computer science in general (Collberg et al., 2016). Reproducibility and replicability have been researched for sometime in Information Retrieval (IR) since the Grid@CLEF pilot track (Ferro et al., 2009). The aim was to create a 'grid of points' where a point defined the performance of a particular IR system using certain pre-processing techniques on a defined dataset. Louridas et al. (2012) looked at reproducibility in Software Engineering after trying to replicate another author's results and concluded with a list of requirements for papers to be reproducible: (a) All data related to the paper, (b) All code required to reproduce the paper, and (c) Documentation for the code and data. Fokkens et al. (2013) looked at reproducibility in WordNet similarity and Named Entity Recognition, finding five key aspects that cause experimental variation and therefore need to be clearly stated: (a) pre-processing, (b) experimental setup, (c) versioning, (d) system output, and (e) system variation. In Twitter sentiment analysis, Sygkounas et al. (2016) stated the need for using the same

software library versions and datasets when replicating work.

Different methods of releasing datasets and code have been suggested. Ferro et al. (2009) defined a framework (CIRCO) that enforces a pre-processing pipeline where data can be extracted at each stage therefore facilitating a validation step. They stated a mechanism for storing results, dataset and pre-processed data[6]. Louridas et al. (2012) suggested the use of a virtual machine alongside papers to bundle the data and code together, while most state the advantages of releasing source code (Fokkens et al., 2013; Potthast et al., 2016; Sygkounas et al., 2016).

Within the the research community, conferences have added reproducible research as a research track, and this started within IR in 2015[7]. These tracks have now progressed into NLP with COLING 2018[8], LREC 2018[9], LREC 2020[10], and COLING 2020[11]. Furthermore, LREC 2020 also had the first shared task on reproducing a set of NLP papers (Branco et al., 2020). Predating these reproducible research tracks at NLP conferences was the 4REAL workshop[12] which encouraged researchers to present and generate reproducible NLP research.

Recently within the NLP and ML fields there has been a growing consensus around what is required to make a paper reproducbilble, with the EMNLP 2020 conference using a checklist derived from Dodge et al. (2019) and the NeurIPS 2020 conference checklist by Joelle Pineau[13]. There has also been a push for better reporting of methods, which relates to the concept of generalisation as well as reproducibility within this thesis. A model card (Mitchell et al., 2019) is one form of reporting a method whereby the creator(s) of the method would state various details of the method including ethical information, method details, and evaluative results. Companies have also created tools to make machine learning more reproducible such as Weights & Biases[14] 'Artifacts'[15] that allows users to track the data and code that created a specific model.

More closely related to TDSA, Marrese-Taylor et al. (2017b) attempted to reproduce three different syntactic target extraction methods. They found that parameter tuning was very important, however using different pre-processing pipelines such as Stanford's CoreNLP did not have a consistent effect on the results. They found that the methods stated in the original papers are not detailed enough to reproduce the study as evidenced by their large results differential. Within sentiment classification at the document level Dashtipour et al. (2016) reproduced numerous non Neural Network (NN) multilingual methods and applied them all to the same two datasets so that the methods can be easily compared. At the sentence level Barnes et al. (2017) compares seven NN and non-NN approaches on six datasets, where many of these datasets have only been benchmarked against one approach. They found a bi-directional LSTM to be on average the most effective approach, and further found that using pre-trained sentiment embeddings from lexically similar data as the training data greatly improves results. Lastly within TDSA,

[6] http://direct.dei.unipd.it/
[7] http://ecir2015.ifs.tuwien.ac.at/wp/?page_id=227
[8] https://coling2018.org/index.html%3Fp=491.html
[9] http://lrec2018.lrec-conf.org/en/calls-papers/1st-call-papers/
[10] https://lrec2020.lrec-conf.org/en/calls-papers/1st-call-papers/
[11] https://coling2020.org/pages/call_for_papers
[12] http://4real.di.fc.ul.pt/
[13] https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf
[14] https://www.wandb.com/
[15] https://www.wandb.com/articles/announcing-artifacts

Chen et al. (2017) reproduced multiple methods and created their own methods, which they then applied to four datasets, which contain two different languages, different domains, and types, but all come from the same medium, written text. Thus, this chapter presents the first reproduction study within TDSA. Also using the methods reproduced that are known to reflect the original paper's implementation, the largest to date study of generalisation within TDSA within the English language that spans six datasets with different types, domains, and mediums.

3.3 Methods

In this chapter, three popular TDSA methods are reproduced, two NP; Vo et al. (2015) and Wang et al. (2017a), and one LSTM Tang et al. (2016b). The LSTM based method was chosen due to the disagreement within the literature, where different prior works have reported different results to the original. The other two methods were chosen as they are fairly different to the LSTM whereby they use word embeddings, but those word embeddings are only input into an SVM compared to a large parameterised LSTM. Secondly within the two NP methods Wang et al. (2017a) is a direct extension of Vo et al. (2015) whereby they include syntactic information directly into the model thus requiring a dependency parser. Therefore all three methods are fairly different. These differences are important in evaluating how generalisable the methods are across varying datasets as stated in the introduction 3.1, as these differences could explain why one method is better than another on certain datasets. Furthermore, even though all three have been compared within Wang et al. (2017a), they have only been compared on Twitter based datasets and not a range of non-Twitter datasets. In this subsection the three methods will be explained in detail, in the following order: Vo et al. (2015) NP method, Wang et al. (2017a) NP with dependency parsing, and Tang et al. (2016b) LSTM.

3.3.1 Neural Pooling

The Vo et al. (2015) NP method is the simplest of the three methods being presented within the chapter. As shown in figure 3.1 it treats each word in the sentence as a word vector that has come from a pre-trained word embedding model, e.g. GloVe (Pennington et al., 2014). From this the method splits the sentence into four different contexts based on the target word(s) position:

1. The whole context – the whole sentence.
2. The left context – all words left of the target word but not including the target word.
3. The target context – the target word(s) (can be more than word e.g. camera lens).
4. The right context – all words right of the target word but not including the target word.

From these variable length contexts, numerous NP methods are applied to these contexts, to create fixed length feature vectors from a variable length sentence. An NP method as stated takes a variable number of word vectors e.g. $W \in \mathbb{R}^{d \times n}$ where n is the

number of words and d is the dimension of the word vector which represent features of the word. The NP then applies a pooling method e.g. maximum value across all the d dimensions of the n words to output a feature vector of $w \in \mathbb{R}^d$ in the example. For each context and NP method a feature vector is created, all feature vectors are then concatenated and are inputted into the linear SVM to classify the sentiment of the target.

The method described above is the general approach Vo et al. (2015) took. More specifically they created four different methods which all used the same NP method but different contexts and one also incorporated sentiment lexicons in a novel way, these are described below:

1. Target-Independent (TI) – Only used the whole context.
2. Target-Dependent Minus (TDM) – Left, right, and target contexts.
3. Target-Dependent (TD) – Left, right, target, and whole contexts (Union of the first two methods).
4. Target-Dependent Plus (TDP) – This incorporated sentiment lexicons by filtering all words that are not in the sentiment lexicon from a given context. This sentiment filtering was applied to the left and right contexts denoted as LS and RS respectively. In total this method used the left, right, target, whole, LS, and RS contexts.

As can be seen above each method from the top of the list incorporates more context or external information as you move down the list of methods. Furthermore from the methods above only the first method will always produce the same sentiment no matter the target word if all targets come from the same sentence, as it does not incorporate any target information.

This method had several important contributions; first is the splitting up of the sentences into different contexts to model simple interaction between the different contexts. Second the use of sentiment lexicons to filter words when the words are represented as word vectors within sentiment analysis. Lastly the extension of NP methods from Tang et al. (2014) original *max*, *min*, and *avg* functions to those listed below with the contribution of also explaining what the functions capture with regards to sentiment:

1. *max* - Maximum value represents positive sentiment.
2. *min* - Minimum value represents negative sentiment.
3. *avg* - Average value represents the average sentiment.
4. *std* - Standard deviation represents the sentiment variation.
5. *pro* - Product represents the average sentiment but with larger differences between positive and negative sentiment.

Overall the general architecture of Vo et al. (2015) can then be summarised through figure 3.1.

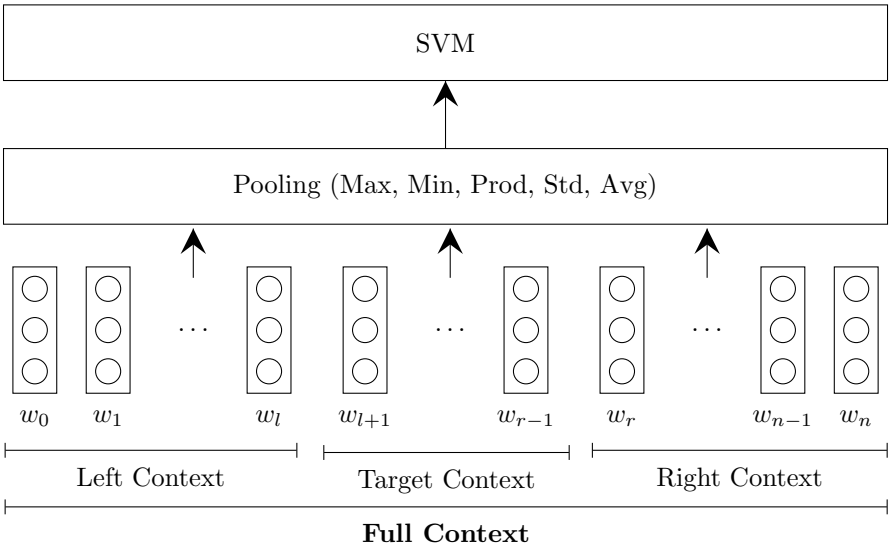


Figure 3.1: General architecture of Vo et al. (2015) and Wang et al. (2017a).

3.3.2 Neural Pooling with Dependency Parsing

The Wang et al. (2017a) method is a direct extension of Vo et al. (2015) where they keep the same NP methods and general architecture but change the contexts. The main motivation of Wang et al. (2017a) is to improve the simple interaction of the different contexts by incorporating the syntactic structure of the target word using a dependency parser. In more detail this dependency context contains the target word and all connected words within the same root. This makes the assumption that the dependency parser that is used can create multiple roots from one text as it assumes the text is made of more than one sentence. If the dependency parser does not create more than one root in a text then the dependency context will be the same as the whole context[16] from Vo et al. (2015) method. The dependency parser they therefore used was the TweeboParser (Kong et al., 2014)[17] which was created for noisy text such as Tweets and hence why multiple roots can occur in one text. This parser was also used because it was especially created for noisy text and the datasets that Wang et al. (2017a) applied this method to were Twitter datasets. Thus one expects that this method will work best on social media type of texts as this method has been developed with this bias.

As with Vo et al. (2015), Wang et al. (2017a) had three different models each containing either more contexts or external information, these are described in more detail below:

1. TDParse Minus – Only used the dependency context.
2. TDParse – Left, right, target, and dependency contexts.

[16]More precisely it will be the same as the whole context with the target word(s) removed, so very similar to the whole context. A python notebook demonstrating the fact that the authors used the dependency parser (TweeboParser) like this and if another parser is used e.g. Stanford the dependency context will be similar to the whole context can be found here: `https://github.com/apmoore1/Bella/blob/master/notebooks/tdparse_parser.ipynb`.

[17]To the author's knowledge it is believed TweeboParser is the only dependency parser that creates multiple roots for a given text. However the author is not an expert within the field of dependency parsing.

3. TDParse Plus – Left, right, target, dependency, *LS*, and *RS* contexts.

Overall the general architecture of Wang et al. (2017a) is in essence the same as that of Vo et al. (2015), thus can then be summarised through figure 3.1.

3.3.3 LSTM

Unlike the previous two methods this is a NN method, and it was the first for TDSA to use an LSTM. Tang et al. (2016b) created three different methods:

1. Standard LSTM (LSTM).

2. Target Dependent LSTM (TDLSTM).

3. Target Connected LSTM (TCLSTM).

The LSTM method is the most basic approach and can be seen in full in figure 3.2. It treats each word within the text as a word vector where the word vector can be either randomly initialised or initialised from a pre-trained word embedding like GloVe. The word vectors are then inputted into the LSTM NN which takes the word vectors from the left most word first to the last word in the sentence. The output from the LSTM at the last word in the sentence (h_n) is fed into a linear layer with a softmax activation function to generate the probability of the sentiment of the text. This approach takes no target context into account, therefore it represents a sentence level sentiment classifier, and was used as the baseline method.

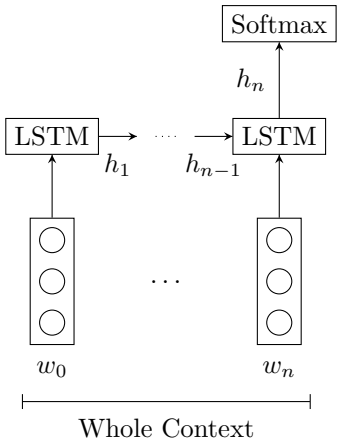


Figure 3.2: Architecture of the LSTM method.

The TDLSTM which can be seen in figure 3.3, is a target specific model, it splits the sentence into two contexts:

1. Left context – The words left of the target word, including the target itself.

2. Right context – The words right of the target word, including the target itself.

Each context has its own LSTM, the left has an LSTM that takes words vectors from left to right, and the right LSTM takes words vectors from right to left. The last word vector(s) that are input into both LSTMs are the target words, this was so that the LSTM could better model the sentiment of the sentence with regard to the target word(s). The final output of both LSTMs (h_{r-1}, h_{l+1}) are concatenated together, which is fed into a linear layer with a softmax activation function to generate the probability of the sentiment of the text.

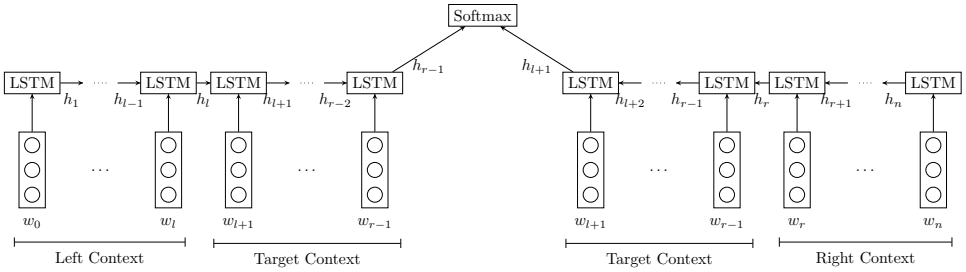


Figure 3.3: Architecture of the TDLSTM method.

Finally the TCLSTM method which can be seen in figure 3.4 is a direct extension of TDLSTM with only one minor difference. The difference is the concatenation of the target word vector (t) to each word vector within the text. The target word vector is represented as the average of all the target word vectors when the target word is made up of more than one word.

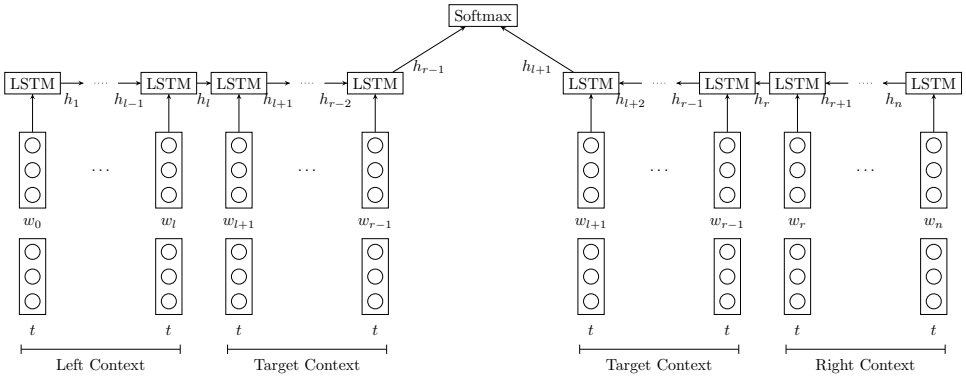


Figure 3.4: Architecture of the TCLSTM method.

3.4 Experimental Setup

3.4.1 Datasets

Within the reproduction and mass evaluation experiments the following datasets will be used:

1. Laptop – SemEval 2014 laptop dataset (Pontiki et al., 2014).
2. Restaurant – SemEval 2014 restaurant dataset (Pontiki et al., 2014).

3. Mitchell – English Twitter dataset (Mitchell et al., 2013)[18].

4. Dong – Twitter dataset (Dong et al., 2014).

5. Election – Election Twitter dataset (Wang et al., 2017a).

6. YouTuBean – Captions from YouTube review videos (Marrese-Taylor et al., 2017a).

Table 3.3 provides a description of the datasets, of which the YouTuBean is by far the smallest dataset. The sentiment class and Distinct Sentiment (*DS*) distributions can be seen in table 3.4. The *DS* (Wang et al., 2017a) is based around the number of unique sentiments per text: DS_1, DS_2, and DS_3 would be all the samples that contain only one, two, and three sentiments in the text, respectively. The idea behind showing the *DS* is a way of judging potentially how difficult a dataset is going to be where the larger i in DS_i the more difficult the dataset might be[19]. Thus from both tables it is clear to see that the datasets are all very different with some being heavily unbalanced with respect to sentiment class distribution, like the Mitchell dataset.

Dataset	DO	T	M	No. Targets	ATS	Uniq	AVG Len
Laptop	L	RE	W	2951	1.58	1295	18.57
Restaurant	R	RE	W	4722	1.83	1630	17.25
Mitchell	G	S	W	3288	1.22	2507	18.02
Dong	G	S	W	6940	1.00	145	17.37
Election	P	S	W	11899	2.94	2190	21.68
YouTuBean	MP	RE/S	SP	798	2.07	522	22.53
DO=Domain, T=Type, M=Medium, ATS=Average targets per sentence, Uniq=No. unique targets, AVG len=Average sentence length per target, L=Laptop, RE=Review, W=Written, R=Restaurant, G=General, S=Social Media, P=Politics, MP=Mobile Phones, SP=Spoken							

Table 3.3: Dataset descriptions and size statistics.

The Dong dataset[20] is a special case out of all of the datasets, as even though it has been used in previous research as shown in table 3.2, it is not a true TDSA dataset. The dataset is more of an Aspect Based Sentiment Analysis dataset as the annotation gives the target and all of its occurrences in the text, rather than just the relevant target occurrence. Example 23 is from the training dataset, where T represents all of the places the target occurs in the text, and here it is clear that the first target (T) is not the target that is negatively affected rather it is the second target (T). This issue of not knowing which is the relevant target was denoted as "same target multiple appearances" by Wang et al. (2017a). Thus when using the Dong dataset for the Tang et al. (2016b) methods the first target (T) is assumed to be the relevant target in all cases, and when using the NP approaches the feature vector of all target occurrences are median pooled

[18]The reason for specifying English is due to Mitchell et al. (2013) also releasing a Spanish Twitter dataset within the same paper.

[19]In chapter 4 it is shown empirically that the large i is within DS_i the more difficult it will be to classify the samples within that distribution.

[20]This dataset can be downloaded from `http://goo.gl/5Enpu7`.

Dataset	Pos (%)	Neu (%)	Neg (%)	DS_1	DS_2	DS_3
Laptop	1328 (45)	629 (21.3)	994 (33.7)	81%	18%	1%
Restaurant	2892 (61.3)	829 (17.6)	1001 (21.2)	75%	23%	2%
Mitchell	707 (21.5)	2306 (70.1)	275 (8.4)	91%	9%	0%
Dong	1734 (25)	3473 (50)	1733 (25)	100%	0%	0%
Election	1744 (14.7)	4572 (38.4)	5583 (46.9)	44%	47%	9%
YouTuBean	224 (28.1)	504 (63.2)	70 (8.8)	82%	18%	0%
Pos=Number of positive targets, Neu=Number of neutral targets, Neg=Number of negative targets, DS_1=1 Distinct Sentiment per sentence, DS_2=2 Distinct Sentiments per sentence, DS_3=3 Distinct Sentiments per sentence						

Table 3.4: Dataset sentiment class and distinct sentiment distributions.

which is the approach used by Wang et al. (2017a). It is not clear how either Vo et al. (2015) or Tang et al. (2016b) handled the 'same target multiple appearance' issue.

T has brought back the female rapper . - really ? T is the biggest parody in popular music since the Lonely Island .

Example 23: An example from the training dataset of Dong (Dong et al., 2014), where T is a placeholder for the target 'nicki minaj'. The sentiment towards the target is negative.

3.4.2 Significance Testing and Evaluation Metrics

As stated earlier, to test if the method has been successfully reproduced the results from the reproduced method will not be statistically significantly different to the original method's results. To choose an appropriate statistical test, the guide from Dror et al. (2018) has been followed, from which the non-parametric paired bootstrap test (Efron et al., 1994) has been selected. This was chosen as the distribution of results cannot always be assumed to come from a normal distribution, due to the macro F1 metric (Dror et al., 2018) which is one of the metrics used within the experiments. Thus this breaks the assumptions of the more powerful parametric tests (student's t-test). Furthermore, out of the two families of non-parametric tests the sampling-based tests are more powerful (Dror et al., 2018; Søgaard et al., 2014) (fewer type 2 errors[21]), from this either Pitman's permutation test or the paired bootstrap can be used, thus we follow Søgaard et al. (2014) in using the paired bootstrap test. The one assumption that has to be made with the paired bootstrap is that the test data is representative of the overall population by having a large enough test set size. Therefore we take the suggestion from Søgaard et al. (2014) and assume that a test set of size greater than 200 is enough to ensure the type 1 errors[22] are minimised. The only experiments that cannot assume the test size is greater than 200 are those that apply five-fold cross validation on the YouTuBean dataset.

[21]Type 2 error: 'refers to the case where the null hypothesis is not rejected although it should be.'(Dror et al., 2018)

[22]Type 1 error: 'refers to the case where the null hypothesis is rejected when it is actually true.'(Dror et al., 2018)

The metrics used in all the experiments will be both the accuracy and macro F1 scores as these are the most frequently used metrics within TDSA (Dong et al., 2014; Wang et al., 2017a; He et al., 2018b). The accuracy score can be calculated using equation 3.1 where TP refers to the number of correctly labelled samples and N are the number of samples in the dataset. The macro F1 score can be calculated using equation 3.3 where the $F1(pos)$, $F1(neu)$, and $F1(neg)$ is the F1 score for the positive, neutral, and negative classes respectively. One of the differences between the two is that the accuracy score is performed globally and thus does not discriminate based on the class label, whereas the macro F1 does as it has to calculate the F1 score for each class and then perform an un-weighted average across all three classes. Due to the un-balanced nature of class labels within all of the TDSA datasets, as shown in table 3.4, the macro F1 score is more biased towards methods that perform well across all sentiments rather than just the dominant sentiment. In comparison, the accuracy score can still be very high with just predicting the most dominant sentiment all of the time. Finally, the majority baseline for accuracy will always be greater than that of the macro F1 due to the un-weighted averaging over the sentiment classes. Thus in general the macro F1 score is a harder metric. The two metrics allow for fair comparison with other works in the community.

$$Accuracy = \frac{TP}{N} \tag{3.1}$$

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{3.2}$$

$$Macro\,F1 = \frac{F1(pos) \; + \; F1(neu) \; + \; F1(neg)}{3} \tag{3.3}$$

The null hypothesis that is being tested within this reproduction section is the following:

Hypothesis 1 *The reproduced method (R) is no better or worse than the original method (O) on a given population* x[23]*.*

While the alternative is:

Hypothesis 2 *The reproduced method (R) is better or worse than the original method (O) on a given population x.*

The null hypothesis 1 will be tested using a two tailed test as it is to be seen if the reproduced method is better or worse rather than just one of those options otherwise a one tailed test would be used. This null hypothesis can be expressed through equations 3.4[24] and 3.5, where $\delta(x)$ is expressed in equation 3.6 and M represents an evaluation metric of which this would be either accuracy or macro F1.

$$p(\delta(X^{25}) \leq 0) \tag{3.4}$$

$$p(\delta(X) \geq 0) \tag{3.5}$$

[23] x is defined later on in this section.
[24] X is defined later on in this section.
[25] X is defined later on in this section.

$$\delta(x) = M(R(x)) - M(O(x)) \tag{3.6}$$

To reject the null hypothesis 1 and accept the alternative hypothesis 2 equation 3.7 or 3.8 has to be true for some given α.

$$p(\delta(X) \leq 0) \leq \frac{\alpha}{2} \tag{3.7}$$

$$p(\delta(X) \geq 0) \leq \frac{\alpha}{2} \tag{3.8}$$

The α value[26] determines the upper bound on the number of type 1 errors that can occur (Dror et al., 2018). Therefore in this thesis an α value of 0.05 is chosen as this is the most popular value used within the field (Liu et al., 2017; He et al., 2018b).

The population x is normally our test set when comparing two systems, and we assume that this test has come from a large population X. The null hypothesis is used to ensure that our one test score on the given test set x has not come about by chance. Therefore to compute the p-value for equations 3.4 and 3.5 we approximate the larger population X using the bootstrap re-sampling method. This is a method of creating Z test sets of size n, which is the same size as the original test set, by sampling from the original test set with replacement. The assumption is that this creates new test sets that are representative but different from the original, which can be used as a proxy of the larger population X. The paired bootstrap procedure is presented in algorithm 1, which is used to approximate the p-values of which p_0 and p_1 can be substituted as the p-value in equations 3.7 and 3.8. For all experiments Z is at least equal to 1000[27], which is the same as Koehn (2004) used and is computationally feasible.

Algorithm 1: The paired bootstrap algorithm adapted from figure 1 in Berg-Kirkpatrick et al. (2012)

Draw Z bootstrap samples of size n by sampling with replacements from x;
$w = 0$;
$b = 0$;
for *each* $x^{(i)}$ *increment* **do**
    if $\delta(x^{(i)}) < 0$ **then**
        $w = w + 1$;
    if $\delta(x^{(i)}) > 0$ **then**
        $b = b + 1$;
end
$p_0 \approx 1 - \frac{w}{Z}$;
$p_1 \approx 1 - \frac{b}{Z}$;
return p_0 and p_1;

This bootstrapping approach can therefore be used to create a confidence range for a method whereby using these Z metric scores as a distribution of results. From this

[26] It is suggested in Søgaard et al. (2014) that an α of 0.0025 should be used and that p-values are recorded.

[27] In some cases where it was computationally feasible Z was 10,000.

distribution removing the top and bottom 2.5% of scores creates the confidence range for an $\alpha = 0.05$[28], for the two tailed test, as shown by Koehn (2004). If the reproduced confidence range does not include the original method's score the null hypothesis must be rejected.

The two tailed test can easily be converted into a one tailed test, to test if a method is better than another, whereby the hypothesis now changes to:

Hypothesis 3 *Reproduced method A is no better than the original method B on a given population x.*

While the alternative is:

Hypothesis 4 *Reproduced method A is better than the original method B on a given population x.*

The null hypothesis 3 can be expressed by equations 3.5 and 3.9.

$$\delta(x) = M(A(x)) - M(B(x)) \tag{3.9}$$

To reject this null hypothesis 3 equation 3.10 has to be true for some α, which in this thesis is 0.05.

$$p(\delta(X) \geq 0) \leq \alpha \tag{3.10}$$

3.4.3 Pre-Processing and Modelling Frameworks

The pre-processing for the two NP approaches will use the Twitter based tokeniser; Twokenizer (Gimpel et al., 2011) as this was stated within Vo et al. (2015) and is shown to be used within Wang et al. (2017a) codebase[29]. For the LSTM approach (Tang et al., 2016b) the English Spacy tokeniser is used due to its speed and wide use within the NLP field, as well as Tang et al. (2016b) not stating the tokeniser used. All text is lower cased after being tokenised.

For the NP methods, scikit-learn's (Pedregosa et al., 2011) LinearSVC is used as it is a wrapper of LibLinear (Fan et al., 2008) which is the library that both NP papers used for their SVMs. For Tang et al. (2016b) LSTM based methods the AllenNLP framework (Gardner et al., 2018) is used which uses PyTorch (Paszke et al., 2019).

3.5 TDSA Reproduction Studies

These studies will explore how the results from the original methods differ with those reproduced. Each paper will have its own subsection detailing the differences and if the differences are statistically significant. Furthermore, at the end of the section based on the results, suggestions are offered to make papers more reproducible. Lastly based on the reproduction results of Tang et al. (2016b) additional suggestions are made for NN based methods to improve reporting of results, which in itself helps with both evaluation and reproducibility. A paper is defined as being reproduced if the main result is not statistically significantly different on at least one metric and if the results have the same

[28] The percentage to remove is equal to $\frac{\alpha}{2} \times 100$ for the two tailed test.

[29] `https://github.com/bluemonk482/tdparse/blob/master/data/dataprocessing.py`

rank order e.g. model A is better than model B for at least one metric. In the thesis, this is how the definition for reproducibility from the introduction 3.1 has been empirically interpreted.

3.5.1 Neural Pooling

To state if the paper (Vo et al., 2015) has been reproduced the last experiment in the paper (table 5 (Vo et al., 2015)) is performed which evaluates the methods on the test set of Dong et al. (2014) Twitter dataset. To be comparable the same SVM C-value parameters are used for all methods and sentiment lexicons for the Target-Dependent Plus method. Also the same word vectors are used which are the concatenation of the unified Sentiment Specific Word Embeddings (SSWE) (Tang et al., 2014) and their own Twitter specific continuous skip-gram (Mikolov et al., 2013b) word vectors (w2v) creating SSWE + w2v. Finally MinMax scaling, as shown in equation 3.11, is used on all features, whereby this enforces a feature to be in a pre-defined *max* to *min* range for all samples ($X \in \mathbb{R}^{n \times d}$ where n is the number of samples and d is the number of features). This therefore enforces all features to be within the same range and "avoid attributes (features) in greater numeric ranges dominating those in smaller numeric ranges" (Hsu et al., 2016). $X_{min} \in \mathbb{R}^d$ and $X_{max} \in \mathbb{R}^d$ are the smallest and largest value respectively for each feature. Within this thesis the *min* and *max* will be 0 and 1 respectively as this is one of the recommended ranges by Hsu et al. (2016). Only after the experiments have been conducted within this thesis, after looking through Wang et al. (2017a) codebase, was it found they used -1 and 1 for *min* and *max*. Thus it will be shown the difference between using these two different *min* and *max* ranges is negligible.

$$X = \frac{X - X_{min}}{X_{max} - X_{min}} * (max - min) + min \tag{3.11}$$

The results from this experiment can be seen in table 3.5 which compares the reproduced to the original. The original did not use the Target-Dependent Minus method within this experiment. As can be seen the results are similar to the original. However by adding only target context information and removing the whole text context (Target-Dependent Minus) does not improve results greatly over the standard non-target aware method (Target-Independent). Further Target-Dependent Minus is not statistically significantly better than Target-Independent on either of the two metrics, as shown by the p-values in table 3.6. Thus showing for the first time that adding the whole text context is statistically significant within the Target-Dependent method and not just on average better, as shown by table 3.6. Figure 3.5 shows the confidence intervals of each reproduced method, and that for the accuracy metric all of the original results are within confidence intervals. Further according to the results all methods are in the same rank order as the original. Thus this paper has been reproduced.

	Accuracy		macro F1	
Model	O	R	O	R
Target Independent	67.3	65.0	66.4	61.9
Target Dependent Minus	0.0	66.6	0.0	62.1
Target Dependent	69.7	69.7	68.0	66.7
Target Dependent Plus	71.1	69.9	69.9	67.6

Table 3.5: Reproduced (R) and original (O) results on the test set of Dong et al. (2014) Twitter dataset.

Metric	Method	TI	TDM	TD	TDP
macro F1	TI	-	0.5459	0.9976	0.9995
	TDM	0.4541	-	0.9995	0.9994
	TD	**0.0024**	**0.0005**	-	0.7606
	TDP	**0.0005**	**0.0006**	0.2394	-
Accuracy	TI	-	0.8245	0.9987	0.9986
	TDM	0.1962	-	0.9935	0.9905
	TD	**0.0017**	**0.0094**	-	0.6284
	TDP	**0.0019**	**0.0121**	0.4223	-

Table 3.6: P-values testing if the methods in the rows are significantly better than the methods in the columns across two metrics. All p-values that are significant ≤ 0.05 are in **bold**.

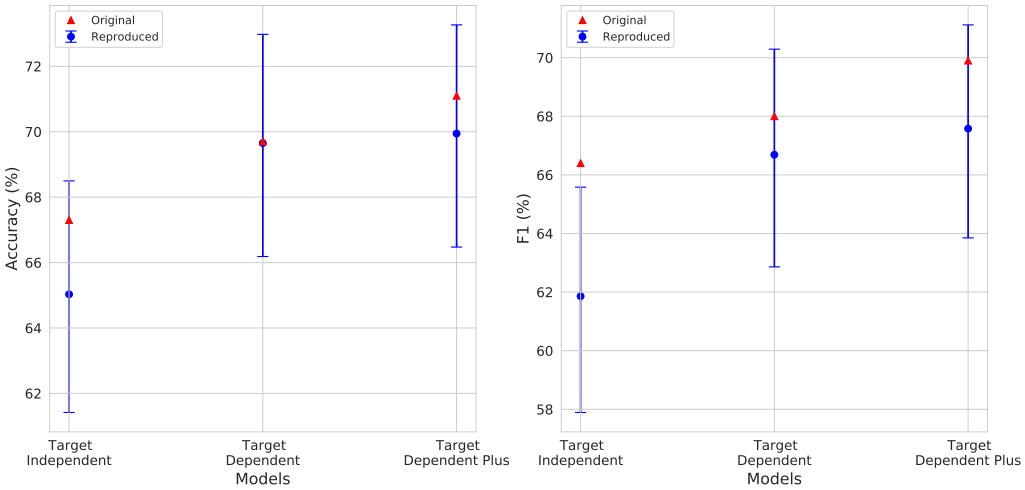


Figure 3.5: Confidence intervals for the two tailed test for the reproduced models of Vo et al. (2015) on both the accuracy and macro F1 metrics.

As stated earlier Wang et al. (2017a) used a different scaling range, -1 to 1, for MinMax scaling. Figure 3.6 shows the confidence intervals for each method when using Wang et al. (2017a) scaling range on the test set of Dong et al. (2014) Twitter dataset. As can be seen the confidence intervals would overlap with those from figure 3.5 which

used the 0 to 1 scaling range that is used throughout the Neural Pooling experiments. Thus showing in this case the scaling range would not appear to be a significant factor.

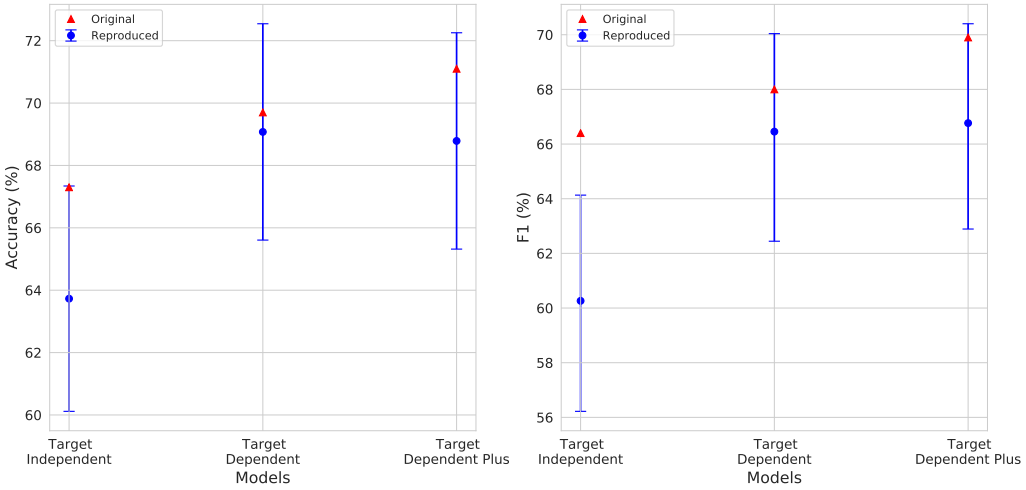


Figure 3.6: Confidence intervals for the two tailed test for the reproduced models of Vo et al. (2015) using Wang et al. (2017a) scaling range of -1 to 1, on both the accuracy and macro F1 metrics.

Given that the paper has been reproduced, further studies are explored. The first of which is comparing different word embeddings, in Vo et al. (2015) they compared w2v, SSWE, and SSWE + w2v. The comparison was done using five-fold cross validation on the training data whereby they report the mean accuracy scores within figure 4 of their paper. This experiment has been recreated, and the word embeddings compared have been expanded to include the non-type non-task specific 300 dimension 840 billion token GloVe embeddings (Pennington et al., 2014) (from now on called GloVe embeddings). These much larger word embeddings are by the far the most popular embeddings within the TDSA literature. Furthermore unlike w2v which are type specific and SSWE which are task and type specific these are neither and more general. Thus it would be of interest to see if general embeddings can perform as well or better than the original task and type specific embeddings. The results of the experiment can be seen in table 3.7. As can be seen the GloVe and the SSWE + w2v are very similar in their performance, and both always outperform the w2v and SSWE embeddings. However unlike the original results, the reproduced results tend to find w2v to perform better than SSWE, as shown by the highlighting in the table.

		Accuracy		macro F1	
Method	Embedding	O	R	O	R
TI	w2v	59.20 (0.00)	60.96 (0.60)	-	56.64 (0.69)
	SSWE	60.70 (0.00)	60.58 (1.08)	-	56.52 (1.46)
	SSWE + w2v	62.30 (0.00)	62.24 (0.91)	-	59.16 (0.66)
	GloVe	-	**63.72** (1.76)	-	**61.31** (1.74)
TDM	w2v	65.40 (0.00)	65.67 (1.11)	-	61.38 (1.29)
	SSWE	66.60 (0.00)	66.74 (0.48)	-	62.77 (0.78)
	SSWE + w2v	67.60 (0.00)	**67.46** (1.04)	-	**64.18** (1.18)
	GloVe	-	67.41 (0.78)	-	64.11 (0.82)
TD	w2v	65.70 (0.00)	66.81 (0.86)	-	62.66 (1.16)
	SSWE	66.70 (0.00)	66.37 (0.59)	-	62.41 (0.81)
	SSWE + w2v	68.30 (0.00)	68.02 (0.82)	-	64.90 (0.91)
	GloVe	-	**68.69** (1.13)	-	**65.68** (1.24)
TDP	w2v	67.40 (0.00)	68.37 (1.17)	-	65.04 (1.39)
	SSWE	67.90 (0.00)	67.72 (1.11)	-	64.39 (1.54)
	SSWE + w2v	69.10 (0.00)	**69.05** (1.19)	-	66.34 (1.41)
	GloVe	-	68.98 (1.09)	-	**66.39** (1.23)

Table 3.7: Mean (standard deviation) metric score for each method and embedding on the Dong et al. (2014) Twitter dataset, where the **bold** value represents the best embedding for each method and metric. Difference in rank order is highlighted.

These results for the embedding comparison so far have only been based on mean accuracy scores from five-fold cross validation. To be more rigorous in evaluation, significance testing is performed whereby the one tailed test is performed on each fold comparing the SSWE + w2v embedding score per metric and method to all other embeddings. The SSWE + w2v embedding was compared to the others as this was the best embedding from the original paper. The p-values generated from these significant tests can be seen in appendix A.1 tables A.1 and A.2 for the accuracy and macro F1 scores respectively. Furthermore as the significance testing is now performed on five folds, which is equivalent to five datasets, thus creating five p-values for each evaluation. Therefore here the number of folds that are significant will be reported. However using the simple approach of counting the number of folds that are less than some α has been shown to

introduce more type 1 errors (Dror et al., 2017) than that was set by the α parameter, which is 0.05, in the individual significance tests. Therefore to stop the introduction of type 1 errors and keep the upper bound to α a correction procedure is required of which Dror et al. (2018) recommends two; Fisher and Bonferroni (Benjamini et al., 2008). The difference between the two is that Fisher should be used when the p-values have come from datasets that are independent, where as Bonferroni can be used for dependent datasets. As each fold does depend on the other folds, the Bonferroni correction will be used here. This thus introduces how significance testing can be performed in general for multiple datasets.

Table 3.8 shows that for at least one of the folds, metric, and methods SSWE + w2v is significantly better than the SSWE and w2v embeddings but not the GloVe. Thus showing like the original paper that SSWE + w2v are the best embedding in general out of SSWE and w2v. Furthermore the GloVe embedding is also tested to see if it is better than the other embeddings using a one sided test and corrected using Bonferroni[30], of which the results can be seen in table 3.9. Thus from both experiments in tables 3.8 and 3.9 it shows that the GloVe embedding is a reasonable replacement for the type and task specific combination of SSWE + w2v.

Method	Embedding	Accuracy	F1
Target Independent	w2v	0	2
	SSWE	1	2
	GloVe	0	0
Target Dependent Minus	w2v	3	4
	SSWE	0	0
	GloVe	0	0
Target Dependent	w2v	0	4
	SSWE	0	2
	GloVe	0	0
Target Dependent Plus	w2v	0	0
	SSWE	1	1
	GloVe	0	0

Table 3.8: The number of folds, out of a possible of five, that the SSWE + w2v embedding is significantly better than the given embedding and method. The significance testing across multiple folds is corrected using Bonferroni.

The last study explores the significance of scaling the features since Vo et al. (2015) never mentions in their paper nor codebase about scaling. So far all results have used MinMax scaling, here the last experiment from Vo et al. (2015) is repeated (table 5 (Vo et al., 2015)) again which evaluates the methods on the test set of Dong et al. (2014) Twitter dataset. In this experiment no scaling is used, results can be seen in figure 3.7, these results can be compared to the other reproduced scaled version of the methods in figures 3.5 and 3.6. None of the non-scaled methods reproduce the results from the original paper, nor do they preserve rank order as the original best performing method (Target Dependent Plus) is now the worst performing method. This shows that scaling is

[30]The p-values associated to these tests can be seen in appendix A.1 tables A.3 and A.4 for the accuracy and macro F1 metrics respectively.

Method	Embedding	Accuracy	F1
Target Independent	w2v	2	5
	SSWE	1	5
	SSWE + w2v	1	1
Target Dependent Minus	w2v	0	1
	SSWE	0	1
	SSWE + w2v	0	0
Target Dependent	w2v	1	4
	SSWE	3	4
	SSWE + w2v	0	0
Target Dependent Plus	w2v	0	0
	SSWE	0	1
	SSWE + w2v	0	0

Table 3.9: The number of folds, out of a possible of five, that the GloVe embedding is significantly better than the given embedding and method. The significance testing across multiple folds is corrected using Bonferroni.

significant for these methods.

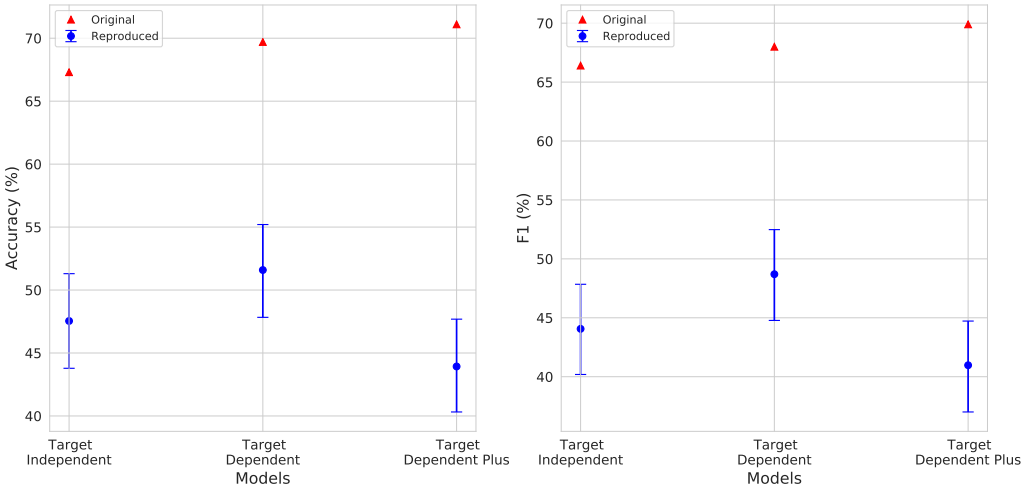


Figure 3.7: Confidence intervals for the two tailed test for the reproduced models of Vo et al. (2015) using no scaling, on both the accuracy and macro F1 metrics. This was evaluated on the test set of Dong et al. (2014).

3.5.2 Neural Pooling with Dependency Parsing

To test if Wang et al. (2017a) methods are reproducible, table 2 and 3 from their paper will be reproduced. These tables test their methods across two Twitter datasets, Dong et al. (2014) and their own Election Twitter dataset. To be comparable the SVM C-value is tuned using five-fold cross validation on the training data, as unlike Vo et al. (2015),

Wang et al. (2017a) does not report what C-value they found to be optimal[31]. The range of C-values used within the tuning process is described within equation 3.12, which is based on the exponential range suggested by Hsu et al. (2016) with the addition of the default C-value (1) for linear SVMs in Scikit-learn (Pedregosa et al., 2011). The best found C-value for the accuracy metric, for each method, on each of the datasets can be seen in table 3.10, these C-values will be used throughout unless otherwise stated. Additionally, the same sentiment lexicons are used, but as stated earlier when using MinMax scaling the features are scaled between 0 and 1 rather than -1 and 1. Also the same word vectors are used, which are the SSWE + w2v. As shown in figures 3.8 and 3.9 the methods have been reproduced on both datasets.

$$C = \{2^n | n = (i \times 2) - 17 \; for \; 0 < i < 10 \; and \; i \in \mathbb{Z}\} \cup \{1\} \tag{3.12}$$

	Methods		
Dataset	TDParse Minus	TDParse	TDParse Plus
Dong	2^{-5}	2^{-7}	2^{-7}
Election	2^{-7}	2^{-9}	2^{-9}

Table 3.10: Best C-values for the accuracy metric for Wang et al. (2017a) reproduced methods.

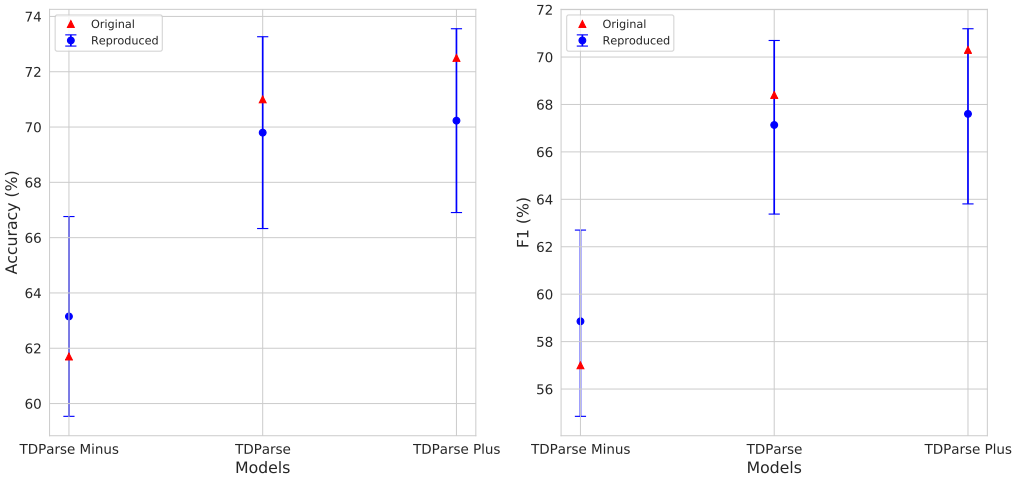


Figure 3.8: Confidence intervals for the two tailed test on the Dong et al. (2014) test set, for the reproduced models of Wang et al. (2017a).

When using the MinMax scale range used by the original paper (Wang et al., 2017a) (-1 to 1) the results are similar for all but the Election macro F1 scores, as shown in figures 3.10 and 3.11. However both the MinMax scale range used in this thesis, and the range used by Wang et al. (2017a) create significantly different results to those of

[31]The range of C-values Wang et al. (2017a) used for tuning was not reported within the paper. However later on after the experiments within the thesis the C-value range used by Wang et al. (2017a) was found through a function in their codebase: `https://github.com/bluemonk482/tdparse/blob/master/src/liblinear.py#L91`.

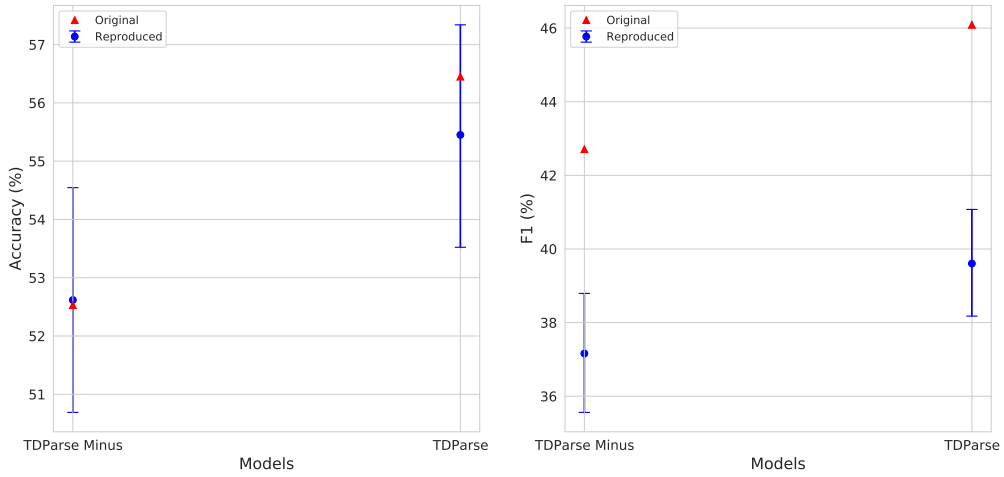


Figure 3.9: Confidence intervals for the two tailed test on the Wang et al. (2017a) Election test set, for the reproduced models of Wang et al. (2017a).

the original paper for the macro F1 metric on the Election dataset. Thus the same experiment is conducted, but using the C-values optimised for the macro F1 metric, where these values can be seen in table 3.11[32]. The results from this experiment can be seen in figures 3.12 and 3.13 for the Election dataset[33] using the scaling range in this thesis and the scaling range of Wang et al. (2017a) respectively. In both cases the macro F1 scores have increased and when using Wang et al. (2017a) scaling range the original scores for macro F1 can be reproduced. Thus showing that within the original paper it is likely that they trained the methods separately with different C-values to optimise the different metric scores.

	Methods		
Dataset	TDParse Minus	TDParse	TDParse Plus
Election	2^{-3}	2^{-7}	2^{-7}

Table 3.11: Best C-values for the macro F1 metric for Wang et al. (2017a) reproduced methods.

As scaling is not mentioned in the paper and only stated within the run command of the codebase, the same experiment is repeated with methods that do not scale the features. As shown in figures 3.14 and 3.15 all results are significantly different and in most cases do not preserve rank order. Thus showing here again the high importance of scaling.

[32]These C-values were tuned using the GloVe embeddings rather than SSWE + w2v. These embeddings were used because the C-values came from the data generated when performing the large scale C-value experiment that is performed later in this section.

[33]The results for the Dong dataset can be seen in appendix A.2 figures A.1 and A.2.

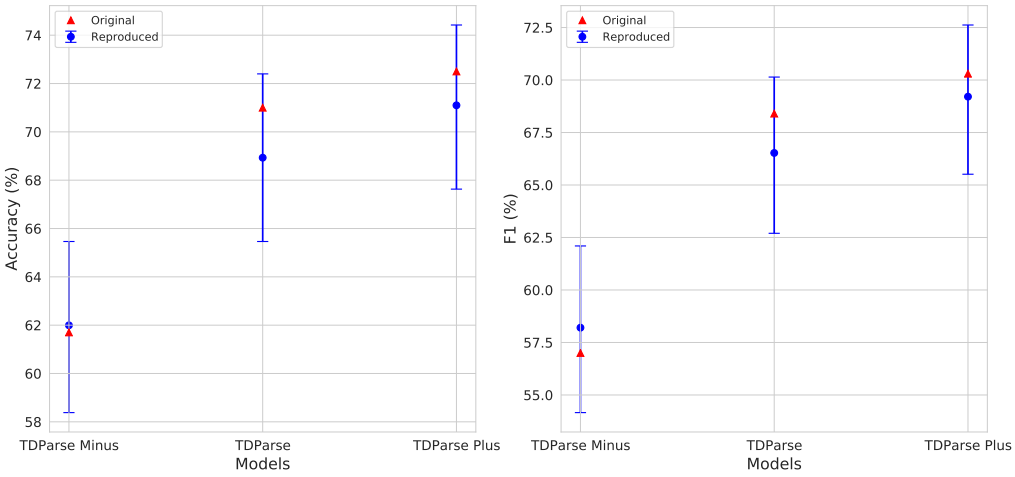


Figure 3.10: Using the original MinMax scaling range of Wang et al. (2017a), the confidence intervals for the two tailed test on the Dong et al. (2014) test set.

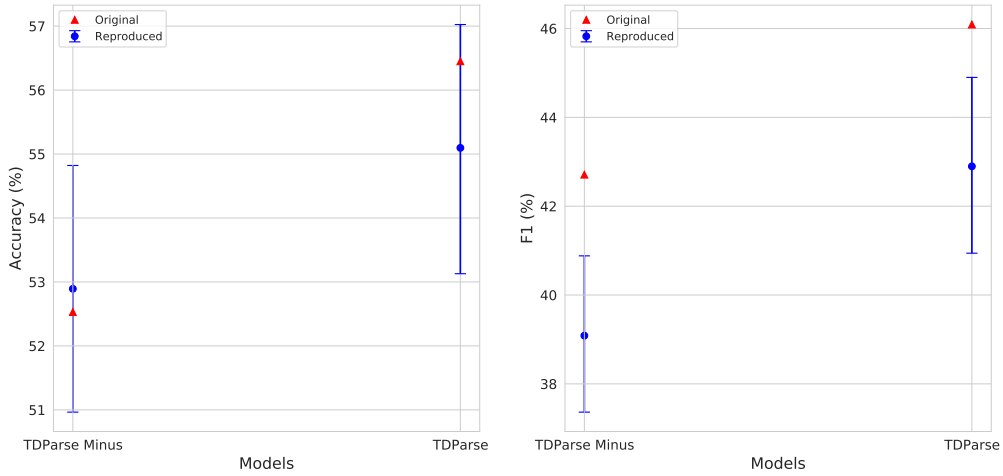


Figure 3.11: Using the original MinMax scaling range of Wang et al. (2017a), the confidence intervals for the two tailed test on the Wang et al. (2017a) Election test set.

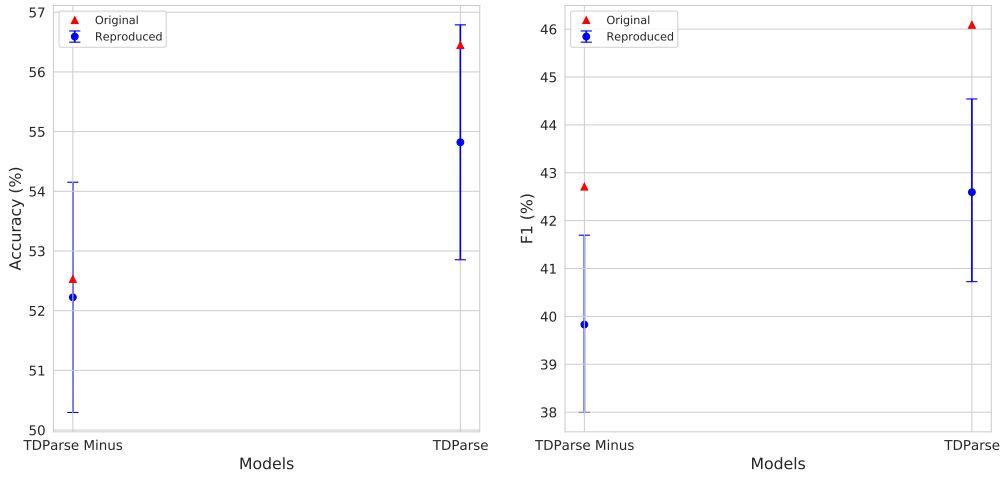


Figure 3.12: Using the C-values optimised for macro F1 metric, the confidence intervals for the two tailed test on the Wang et al. (2017a) Election test set.

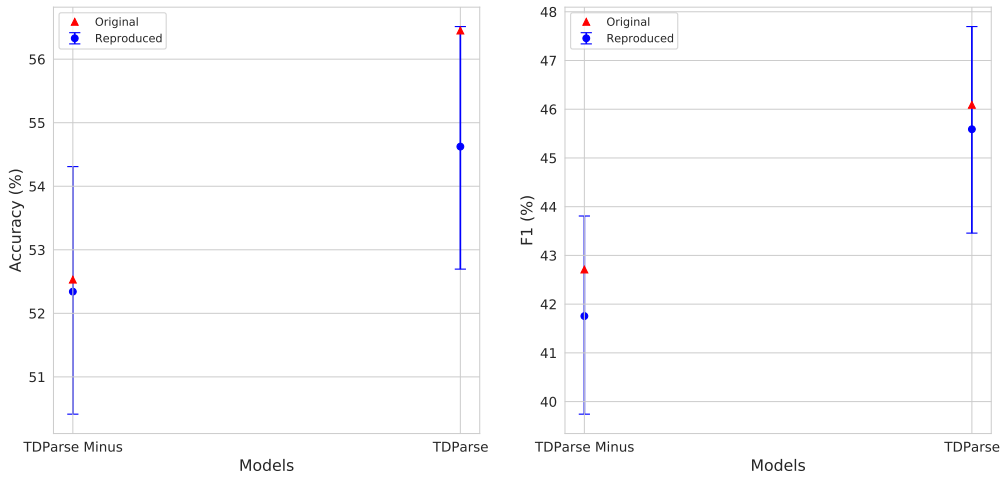


Figure 3.13: Using the C-values optimised for macro F1 metric with the original MinMax scaling range of Wang et al. (2017a), the confidence intervals for the two tailed test on the Wang et al. (2017a) Election test set.

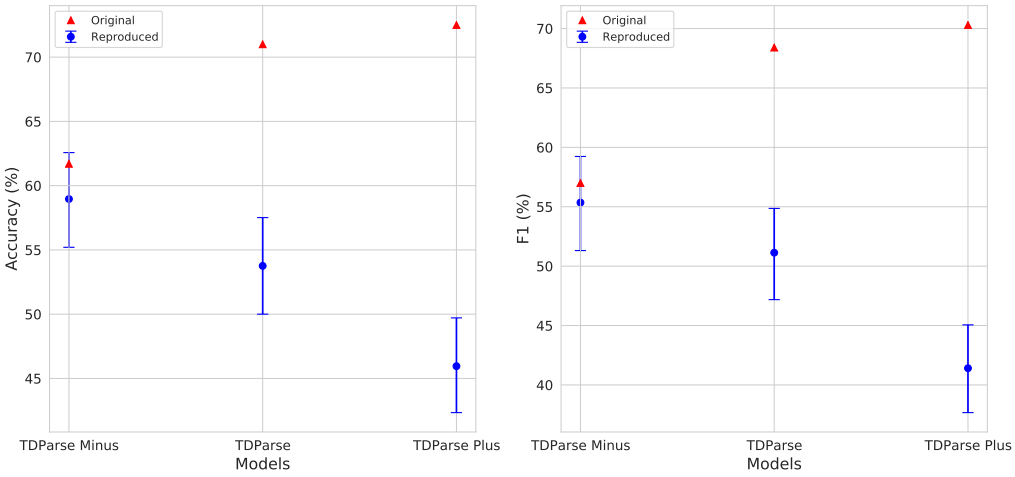


Figure 3.14: Using no scaling, the confidence intervals for the two tailed test on the Dong et al. (2014) test set.

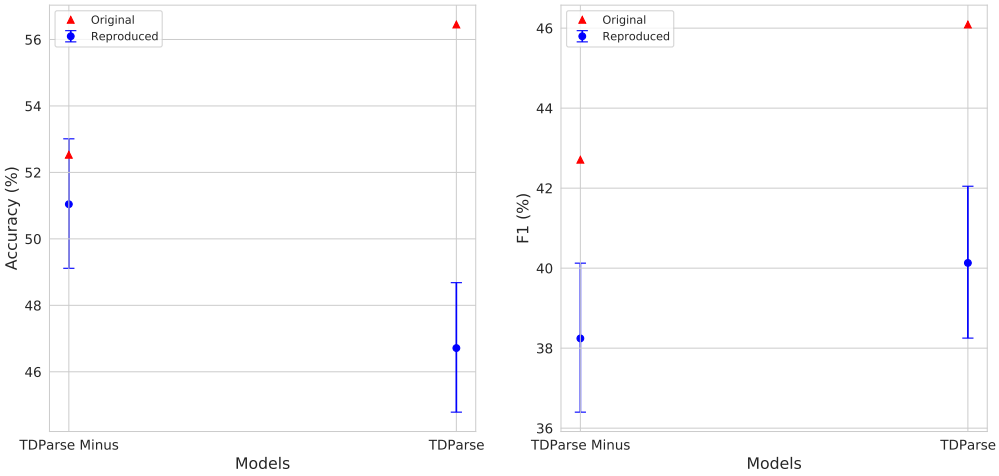


Figure 3.15: Using no scaling, the confidence intervals for the two tailed test on the Wang et al. (2017a) Election test set.

3.5.3 Large Scale Analysis of the Affect of the C-value and Scaling on Neural Pooling Methods

These two sets of experiments will explore the importance of the C-value within the SVM that is used in the NP methods and scaling. As it has been shown that the C-value is statistically significantly important to recreate the macro F1 score within the Election dataset experiments for Wang et al. (2017a) and for all NP methods scaling is statistically significant. These sets of experiments will explore how significant these two parameters are within the NP methods, whereby all of the NP methods from Vo et al. (2015) and Wang et al. (2017a) will be used to make the findings more robust. Furthermore to make the findings generalisable the methods will be applied to all six datasets from table 3.3. The experiments will use five-fold cross validation on the training sets to ensure the test set is not used, and thus allow researche's to use these findings without overfitting to the test sets. In all experiments the mean best performing configuration of the method for either the accuracy or macro F1 metric will be compared against all other configurations using a one sided significant test. The number of folds that the mean best configuration is significantly better than the other configurations will be corrected using Bonferroni.

As these experiments are conducted on various different datasets, the methods will use the general non-type and non-task specific GloVe embeddings that are also the most popular in the area. For sentiment lexicon based methods the same lexicons used by Wang et al. (2017a) will be used as they are a superset of the lexicons used by Vo et al. (2015) and the lexicons come from various types of data[34]. The Stanford CoreNLP tokeniser (Manning et al., 2014) will be used in preference to the Twitter specific tokeniser; Twokenizer (Gimpel et al., 2011) to avoid type specific tools. For Wang et al. (2017a) methods that require a dependency parser the TweeboParser (Kong et al., 2014) will be used on all datasets but the Laptop and Restaurant datasets, whereby the Stanford CoreNLP dependency parser will be used. This decision was made before realising the importance of a dependency parser creating multiple roots for Wang et al. (2017a) methods and Stanford's parser does not create multiple roots. Thus in effect Stanford's parser will create a context that is almost identical to the whole text context[35]. Stanford's parser was chosen due to the Laptop and Restaurant datasets coming from a different type of data (review rather than social media) and thus is believed to require a more type relevant parser, such as the Stanford parser.

The SVM C-value is part of the L2-regularised L2-loss function of the linear SVM which can be seen in equation 3.13 (taken from equation 1 in (Fan et al., 2008)). The L2-regularisation in the equation is $\frac{1}{2}w^T w$ and the rest is the L2-loss. As can be seen the C-value determines the amount of weight the L2-loss has on the overall loss function. Thus if the C-value is large the effect of the regularisation is small, which would more likely cause the model to overfit to the training data.

[34]The lexicons used originate from MPQA (Wilson et al., 2005) (news data), NRC (Mohammad et al., 2010) (general data chosen from a dictionary based on word frequency from Google n-gram corpus (Brants et al., n.d.)), and Hu et al. (2004a) (review data).

[35]The reason it is not the same is due to the context not including the target word(s). Also Stanford's parser requires the text to go through their sentence splitter first, which in some cases does cause the text to be split up.

$$\frac{1}{2}w^T w + C\sum_{i=1}^{l}(\max(0, 1 - y_i w^T x_i))^2 \tag{3.13}$$

The C-values that will be tested here are the same as those evaluated for the reproduction of Wang et al. (2017a), shown in equation 3.12. Figures 3.16 and 3.17 show the mean best C-values for each method on each dataset for the accuracy and macro F1 metric respectively. In both cases it can be seen that all methods are significantly sensitive to the choice in C-value no matter the dataset. In most cases and more so for the accuracy metric the default C-value from scikit-learn (Pedregosa et al., 2011) (1) is significantly worse than the mean best. It would be assumed that datasets that are small like YouTuBean would prefer a small C-value to stop the methods from overfitting, but that does not seem to be the case. Rather it would appear that for both the accuracy and macro F1 score each have their own preferable band of C-values. For accuracy, generally $7.81e^{-3}$ performs well on all methods and datasets, whereas for macro F1 it is dataset and method specific. Furthermore a C-value that performs best for accuracy could be significantly worse than the best C-value for macro F1, of which this happens the most on the Election dataset. For the mean best scores produced by these C-values for all methods on all datasets see figures A.3 and A.4 in appendix A.2.

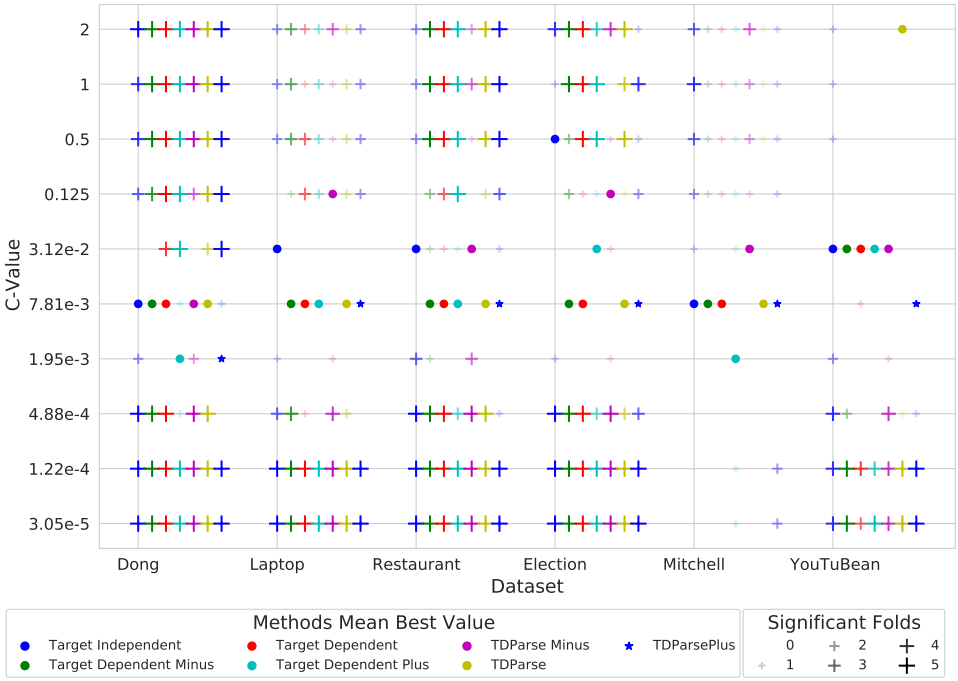


Figure 3.16: For the accuracy metric the mean best C-value for each method and dataset represented by dots and star. The size of the cross indicates the number of folds the mean best C-value is significantly better than the other C-values for the given method and dataset.

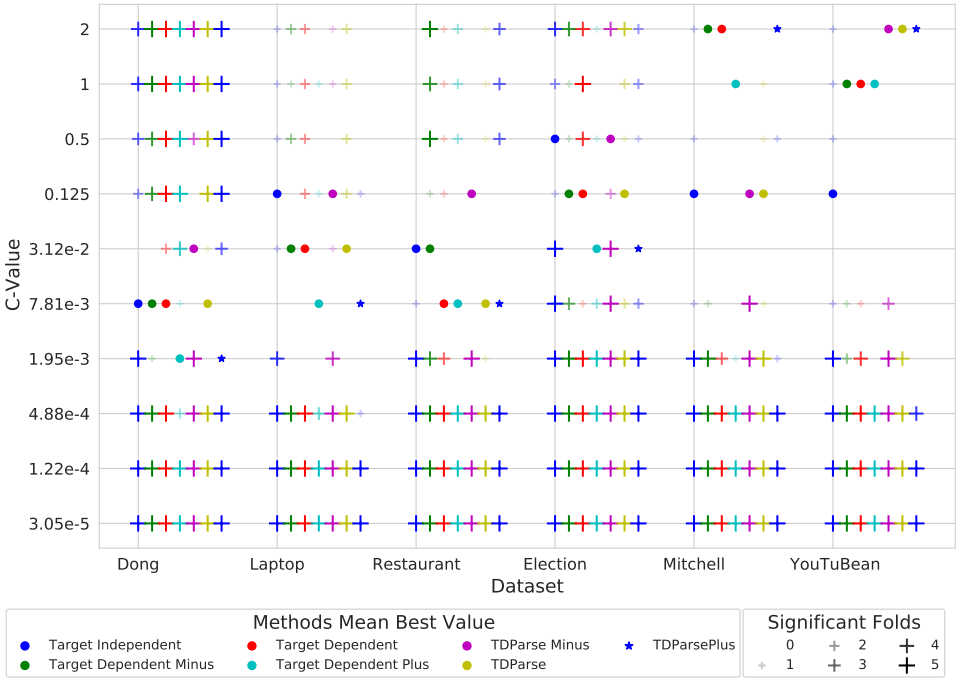


Figure 3.17: For the macro F1 metric the mean best C-value for each method and dataset represented by dots and star. The size of the cross indicates the number of folds the mean best C-value is significantly better than the other C-values for the given method and dataset.

MinMax scaling with the scale range of 0 to 1, rather than Wang et al. (2017a) −1 to 1, will be compared to not scaling. When performing these experiments the optimal C-value for each method on each dataset for the accuracy metric is used, which was found from the last experiment. The results can be seen in figures 3.18 and 3.19. It can be clearly seen in all cases for the accuracy metric and the majority for the macro F1 metric that scaling is statistically significant no matter the method nor dataset, with the caveat of Mitchell for the macro F1 metric. For the mean best scores produced by these scaling experiments for all methods on all datasets see figures A.5 and A.6 in appendix A.2.

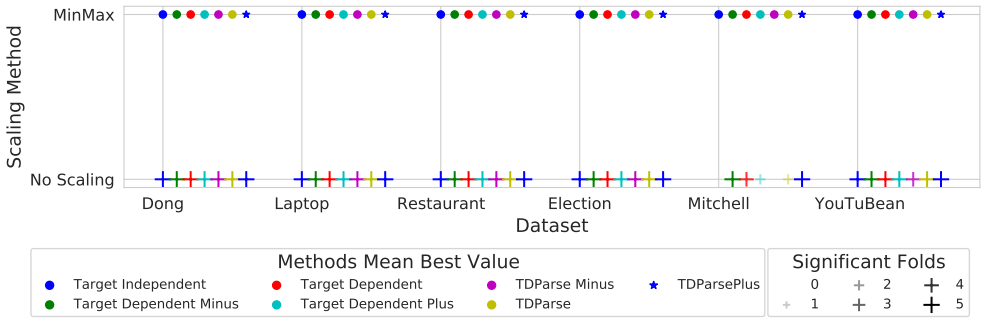


Figure 3.18: For the accuracy metric the mean best scaling method for each method and dataset represented by dots and star. The size of the cross indicates the number of folds the mean best scaling method is significantly better than the other scaling method for the given method and dataset.

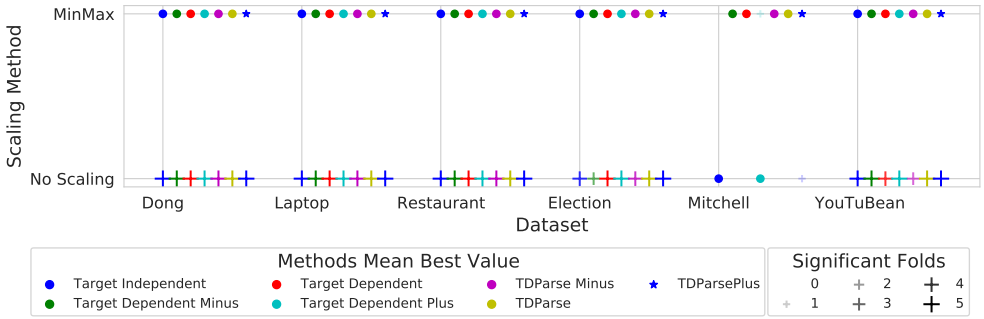


Figure 3.19: For the macro F1 metric the mean best scaling method for each method and dataset represented by dots and star. The size of the cross indicates the number of folds the mean best scaling method is significantly better than the other scaling method for the given method and dataset.

3.5.4 LSTM

It has been found that in the previous work it is difficult to either reproduce (Tay et al., 2018) or replicate (Chen et al., 2017) Tang et al. (2016b) LSTM based methods, most specifically the TDLSTM version, as shown by table 3.12[36]. Based on these findings the

[36]The original Tang et al. (2016b) methods as stated in this section were never originally evaluated on the Laptop or Restaurant datasets. However the original authors within another paper (Tang et al.,

methods in Tang et al. (2016b) are reproduced. The reproduced methods are evaluated on Dong et al. (2014) Twitter dataset and use the same GloVe Twitter 100 dimension embeddings (Pennington et al., 2014). All methods used Stochastic Gradient Descent (SGD) with a learning rate of 0.01, a cross entropy loss, and the hidden dimension of all LSTMs equal to the dimension of the embedding being used. However the paper did not state the number of epochs the method was trained for thus early stopping is used keeping track of the loss value with a patience of 10. As early stopping requires a validation set, the training set is split 80% training and 20% validation. The paper also mentioned that they "set the clipping threshold of softmax layer as 200" (Tang et al., 2016b) as this did not make sense, this was not used. Lastly all weights were initialised using $\mathcal{U}(-\sqrt{k}, \sqrt{k})$[37] where $k = \frac{1}{\text{embedding dimension}}$. The original initialisation from Tang et al. (2016b), $\mathcal{U}(-0.003, 0.003)$, always overfitted to the dominant class[38] when used in the reproduced methods, hence the difference in the initialisation distributions. The tokeniser used was Spacy, as the original tokeniser that was used was not stated in the paper[39].

Authors	Restaurant	Laptop
Tang et al. (2016a)	75.63	68.13
Chen et al. (2017)	78.00	71.83
Tay et al. (2018)	69.73	62.38
□ Original authors	□ Replicated	□ Reproduced

Table 3.12: Accuracy of the TDLSTM method by the different authors on the Restaurant and Laptop datasets.

The results from the experiment can be seen in table 3.13 and figure 3.20 whereby each reproduced method has been ran 20 times. Running each method 20 times using different random seeds allows the methods to take into account the random initialisation problem (Reimers et al., 2017). It can be seen that if the maximum score is used the original and reproduced results are quite close, and statistically similar as shown by figure 3.21[40]. Furthermore based on the maximum score the rank of the methods are the same as the original, thus the methods have been reproduced successfully. However the difference between the maximum result and the minimum can be quite large, especially for the macro F1 metric.

2016a) evaluated the TDLSTM method they created on the Laptop and Restaurant datasets and that is what is meant by original authors within the table.

[37] This is the default initialisation within PyTorch (Paszke et al., 2019) and AllenNLP (Gardner et al., 2018).

[38] Which is neutral for Dong et al. (2014) as shown by table 3.4.

[39] To note in the original paper (Moore et al., 2018) that this chapter is based on the results for Tang et al. (2016b) did use the original weight initialisation of Tang et al. (2016b) ($\mathcal{U}(-0.003, 0.003)$) and could still reproduce the results. The main implementation difference between Moore et al. (2018) and this chapter is that here PyTorch (Paszke et al., 2019) and AllenNLP (Gardner et al., 2018) is used rather than Keras (Chollet et al., 2015) and also the Spacy tokeniser is used rather than Twokenizer Gimpel et al., 2011 tokeniser. It is unknown why the original weight initialisation did not work within the code implementation of this chapter, it is believed it could be due to subtle differences between Keras and PyTorch.

[40] The best run for accuracy is not always the same best run for macro F1. Therefore we only use the best run for each method based on the metric being evaluated.

Metric	Method	Max	Mean	Min	Original
Accuracy	LSTM	64.31	62.93	61.42	66.5
	TDLSTM	69.36	**67.23**	**65.46**	70.8
	TCLSTM	**70.23**	66.74	64.74	71.5
macro F1	LSTM	61.93	58.59	54.43	64.7
	TDLSTM	66.58	**63.86**	59.94	69.0
	TCLSTM	**67.61**	63.26	**60.29**	69.5

Table 3.13: The max, mean, and minimum (min) scores from each reproduced method over 20 runs. The last column are the original scores from the Tang et al. (2016b) paper. The **bold** scores represent the best performing score between the methods for each metric, max, mean, and minimum.

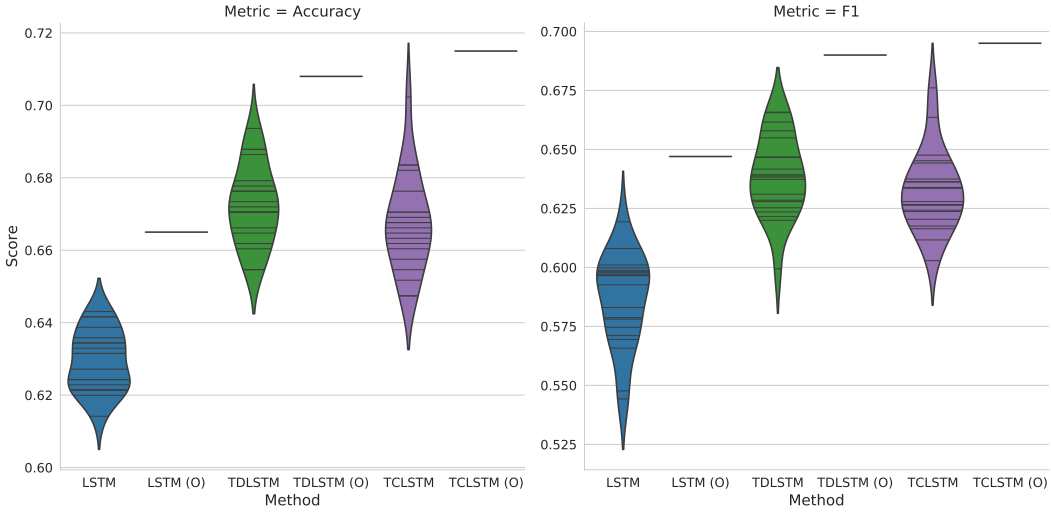


Figure 3.20: The distribution of scores for each of the 20 runs for each reproduced method. Whereby each horizontal line in the distribution represent the result of one run for the reproduced method. Model names with a *(O)* represent the score reported in the original models paper.

Based on the results from table 3.13 it would appear that for many seed values it would not be possible to reproduce these results. To quantify this, the best performing run for each method and metric is compared against all other runs for that method. The number of runs that are significantly worse based on a one sided test corrected using Bonferroni to the best performing run is shown in table 3.14. As can be seen out of the 19 other runs many of them for the macro F1 metric for all methods are significantly different to the best performing run even though they are the same method. This confirms the findings of Reimers et al. (2017) for TDSA for the macro F1 metric whereby random seeds can cause statistically significantly different results for the same method. It further shows that out of all of the methods the TCLSTM would appear to be the least stable, and this could be due to it being the largest, with respect to the number of parameters, of the three methods. From these findings and the distribution of results presented in figure 3.20, it is believed that a possible reason why others could not reproduce (Tay

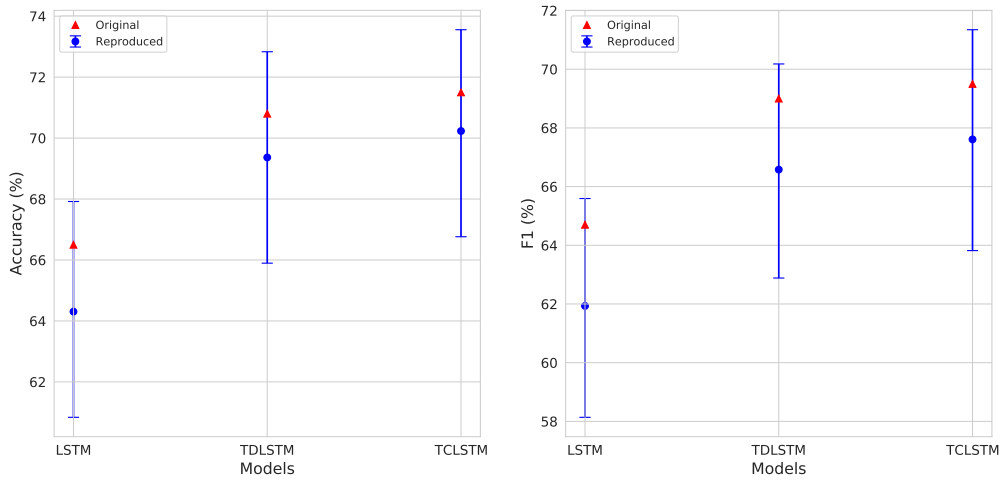


Figure 3.21: Confidence intervals for the two sided tailed test for the reproduced models of Tang et al. (2016b) on both the accuracy and macro F1 metrics.

et al., 2018) or replicate (Chen et al., 2017) the same scores as the original authors of TDLSTM (Tang et al., 2016a) is due to the variance caused by random seeds.

Metric	LSTM	TDLSTM	TCLSTM
Accuracy	0	0	13
F1	4	2	15

Table 3.14: The number of runs that the best performing run significantly outperforms using a one side tested and corrected with Bonferroni for accuracy and macro F1.

As the methods from Tang et al. (2016b) have been reproduced, another of Tang et al. (2016b) experiments is repeated and enhanced. The original experiment compares SSWE, GloVe Twitter 50, 100 and 200 embeddings of which both of these embeddings are either type or type and task specific. Thus the experiment is enhanced to include the GloVe 300 dimension non-task nor type specific embedding, that has been already tested in numerous other experiments in this section (see tables 3.7 and 3.8 for example). The results on the test and validation set of Dong et al. (2014) Twitter dataset can be seen in tables 3.15 and A.5 respectively. The findings are slightly different from the original found in figure 3 of Tang et al. (2016b). In comparison Tang et al. (2016b) found the TDLSTM to be worse than TCLSTM on all accuracy scores, which is not the case here. Additionally Tang et al. (2016b) found SSWE to be worse than GloVe Twitter 50 which again is not found here. These differences in results could be due to the random seeds. More interestingly it is shown that the non-type nor task specific embeddings is better than all other embeddings for all methods and metrics. Thus showing again that these larger general embeddings can be at least as good as the type and/or task specific embeddings. After performing a one tailed test comparing the GloVe 300 embeddings to all other embeddings for each metric and method they are significantly better in the majority of cases for the test set, as shown in table 3.15. The significant results are hard to interpret as the test results suggest that the majority of embeddings for at least

TDLSTM and TCLSTM are significantly worse than the GloVe embeddings, but this is not reflected in the validation results as the GloVe embeddings are not significantly better than any of the embeddings[41]. Thus it suggests that perhaps the GloVe embeddings are only weakly better than all other embeddings[42].

Embedding	Metric	LSTM	TDLSTM	TCLSTM
SSWE	Accuracy	62.07 (1.48)	66.77* (1.63)	65.59* (1.41)
	F1	58.37* (2.16)	63.35* (1.89)	61.96* (1.77)
Twitter 50	Accuracy	61.48* (1.43)	65.11* (1.46)	64.74* (1.87)
	F1	57.12* (2.79)	61.67* (2.00)	60.72* (2.57)
Twitter 100	Accuracy	62.93 (0.81)	67.23* (1.08)	66.74* (1.32)
	F1	58.59 (1.90)	63.86* (1.68)	63.26* (1.68)
Twitter 200	Accuracy	62.49 (1.14)	68.11* (0.54)	67.89 (0.98)
	F1	57.41* (2.81)	65.21* (0.94)	64.70 (1.39)
GloVe 300	Accuracy	**64.73** (0.76)	**71.04** (0.68)	**69.22** (1.27)
	F1	**61.04** (1.45)	**68.43** (0.83)	**66.47** (1.78)

Table 3.15: Test set mean (standard deviation) results on the Dong et al. (2014) Twitter dataset, across various embeddings and methods. The **bold** values indicate the best embedding score for each method and metric. The * indicates when the GloVe embeddings are statistically significantly better ($p \leq 0.05$) than the other embedding for that metric and method. The significance test used the one tailed test and used the median best run from the 20 runs to perform the significance test.

3.5.5 Conclusion from Reproduction Studies

It is clear from the large scale NP experiments shown in figures 3.16 (3.17) and 3.18 (3.19) for the accuracy (macro F1) metric, that both the scaling method (if used) and the C-value from the SVM should be stated within the paper. Furthermore, the suggestion from Reimers et al. (2017) on reporting multiple runs of the method over different random seed values is required for NN based TDSA methods as the single performance scores can be misleading, which could explain why previous papers obtained different results to the original for the TDLSTM method (Chen et al., 2017; Tay et al., 2018). For the first time, it has been shown in this section that scaling method, C-value of the SVM, and random seeds make a significant difference for TDSA methods.

[41]The p-values from the significant tests can be seen in appendix A.1 tables A.6 and A.7 for the test and validation results.

[42]The potential reason for the significance tests to bring back very different results for the test and validation sets could be due to the limitation of the significance test used for the LSTM methods. In this chapter the LSTM based methods when being tested to detect significant differences a particular run from the set of all runs for each compared LSTM method is used. In the majority of cases the median run is used and this to some degree ensures that there is not a bias towards methods that have a large variance in results due to the random seeds. However using the median run or any one run does not take into the account the whole random seed distribution. Thus stating the limitation of the significance test used in this chapter for the LSTM based methods. This limitation could also explain why there is a difference between the significance results of the validation and test set here, as the median run might not be a good representation of all runs results.

Additionally, within this section it has also been shown that general word embeddings (300 dimension GloVe embeddings) can perform as well as the type and/or task specific embeddings. This has been shown for the NP methods (Vo et al., 2015), where as for the LSTM methods (Tang et al., 2016b) they tend to prefer the general GloVe embeddings. From this it shows that smaller type and/or task specific embeddings can be as useful as larger general embeddings for at least the NP methods. This implies that from an energy efficiency perspective[43] it would be of use to train these smaller type and/or task specific embeddings as the methods that use them will be more efficient than if they use the larger general embeddings. Alternatively it suggests that it is not a requirement to create smaller type and/or task specific embeddings. These findings are at least true for TDSA methods applied to the Dong dataset.

3.6 Mass Evaluation[44]

Given the methods from the three reproduced papers, in this section we evaluate the different methods across six different English datasets that are shown in table 3.3. This will be the first TDSA study that has evaluated methods across different types, mediums, and domains, as well as the largest TDSA evaluation with respect to the number of datasets. This study will thus explore whether the three papers' methods perform differently on these various datasets, seeing if any generalise to all datasets. For all methods they will use the same 300 dimension GloVe embedding as it is the most popular and has been shown within the thesis to perform well on data that it was not originally designed for (i.e. social media type data). All methods will use the Spacy tokeniser and the Wang et al. (2017a) methods will use the TweeboParser (Kong et al., 2014) dependency parser on all datasets[45]. For sentiment lexicon based methods, the same lexicons that Wang et al. (2017a) employed will be used here. For the NP methods, the best performing C-value for each method and dataset for the accuracy metric found through the experiment conducted in figure 3.16 will be used. Also MinMax scaling will be used for all NP methods. The LSTM based methods will use the same setup as that from section 3.5.4, but the methods will only be run six times for each dataset due to the computational cost[46]. Further, for all of the LSTM methods the mean result from the six runs will be reported unless otherwise stated. For the NP methods only the two top performing methods are evaluated, again to save on computational cost while still allowing us to compare methods that do and do not use sentiment lexicons.

The accuracy and macro F1 results on the test set for all datasets can be seen in tables 3.16 and 3.17. The statistically significant results comparing each method can be seen for both metrics in figure 3.22, for the LSTM methods the median best performing

[43]Measured by number of parameters. A better measure for efficiency would be Floating Point Operations (FPO) (Schwartz et al., 2019), but Schwartz et al. (2019) did state that the number of parameters is a form of efficiency measure.

[44]Within this section the TDParse Plus and Target Dependent Plus methods will also be called TDParse+ and TD+ respectively.

[45]This is due to the method's requirements of a dependency parser that produces multiple roots.

[46]Six runs was chosen as according to Reimers et al. (2018) as it will allow future researcher's to compare results using significance tests that take all six runs into account. This test was not used in this chapter as neural methods that require running multiple runs are compared to non-neural methods that do not, and there is no literature that is known to the author that states how to compare multiple run performances to single run performances.

run based on the accuracy metric is used to compare to all other methods. From these results it is clear that the NP methods are by far the better set of methods. Additionally it can be seen that adding sentiment lexicons (NP methods with a '+' in their name) only marginally improve results on some datasets and at most significantly better on one dataset. These findings are also similar when comparing the non-dependency parser approaches (TD and TD+ methods) with those that do use a dependency parser (TDParse and TDParse+), whereby they perform almost as well as each other. Both of these findings show, for at least the methods tested here, that using additional resources such as sentiment lexicons or a dependency parser do not make a large significant difference in results and are thus not needed.

	D	E	L	M	R	Y	Mean
LSTM	<u>64.57</u>	<u>47.85</u>	59.90	71.04	<u>68.63</u>	<u>63.33</u>	<u>62.55</u>
TDLSTM	**71.12**	**57.50**	61.76	<u>70.52</u>	73.56	64.17	66.44
TCLSTM	68.98	57.40	<u>56.77</u>	70.77	71.85	66.81	65.43
TD	68.50	57.22	66.14	73.45	77.32	82.50	70.86
TD+	70.23	53.21	**68.97**	**74.37**	78.04	81.67	71.08
TDParse	67.77	57.46	67.08	73.96	77.95	79.58	70.63
TDParse+	69.36	56.12	68.50	73.35	**78.30**	**83.33**	**71.49**
Mean	68.65	55.25	64.16	72.49	75.09	74.48	-
D=Dong, E=Election, L=Laptop, M=Mitchell, R=Restaurant, Y=YouTuBean							

Table 3.16: Accuracy results on the test sets of each dataset. For the LSTM based methods this is the mean accuracy result. The mean accuracy across all datasets for each method is in the right most column. Where the **bold** and <u>underlined</u> values indicate the best and worst methods for each dataset and the overall mean accuracy, respectively. The mean accuracy score for each dataset is in the last row.

	D	E	L	M	R	Y	Mean
LSTM	<u>61.58</u>	<u>30.65</u>	<u>41.91</u>	35.51	<u>37.15</u>	<u>25.85</u>	<u>38.78</u>
TDLSTM	**68.54**	42.54	49.82	<u>29.67</u>	56.66	28.98	46.04
TCLSTM	65.92	43.57	45.04	38.42	53.54	36.83	47.22
TD	65.27	46.60	57.86	48.98	63.17	**74.80**	59.45
TD+	67.36	44.52	**62.33**	48.08	64.44	72.90	59.94
TDParse	64.33	**46.64**	59.23	48.58	64.66	70.05	58.91
TDParse+	66.36	46.30	61.89	**51.17**	**65.26**	74.46	**60.91**
Mean	65.62	42.98	54.01	42.92	57.84	54.84	-
D=Dong, E=Election, L=Laptop, M=Mitchell, R=Restaurant, Y=YouTuBean							

Table 3.17: Macro F1 results on the test sets of each dataset. For the LSTM based methods this is the mean macro F1 result. The mean macro F1 across all datasets for each method is in the right most column. Where the **bold** and <u>underlined</u> values indicate the best and worst methods for each dataset and the overall mean macro F1, respectively. The mean macro F1 score for each dataset is in the last row.

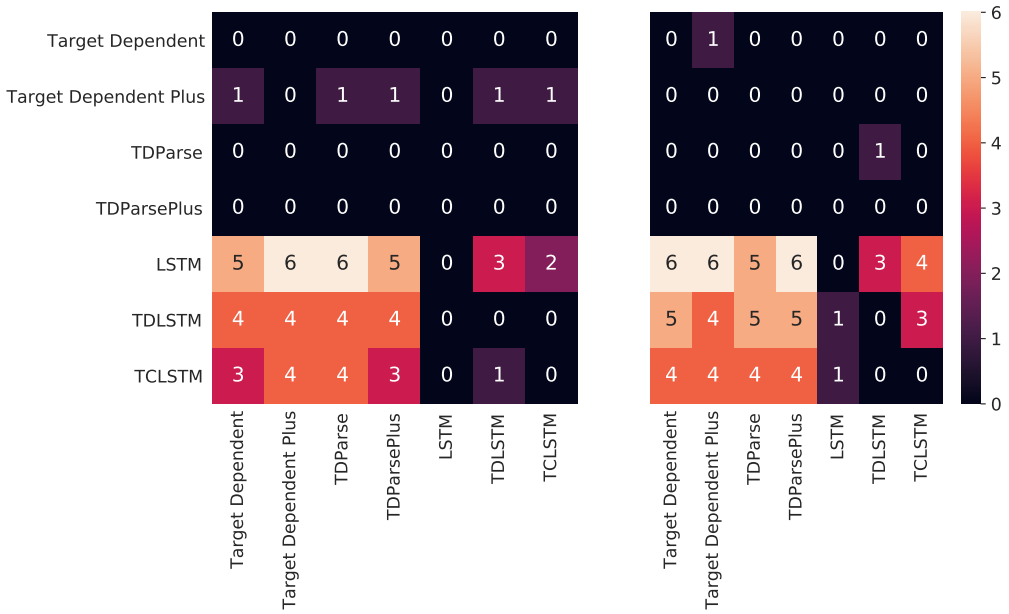


Figure 3.22: The number of datasets where the column methods are statistically significantly better than the row methods, corrected using Bonferroni. The left and right heatmap represent the accuracy and macro F1 scores. This is using the median performing run based on the accuracy metric for the LSTM methods.

Between the LSTM based methods it is clear that on average the target specific LSTM based methods (TDLSTM and TCLSTM) are better than the sentence level classifier baseline (LSTM). However the TDLSTM and TCLSTM are worse than the LSTM on the Mitchell dataset, and the TCLSTM is worse than LSTM on the Laptop dataset when using the accuracy metric. Both of these represent a failure for the target specific methods as they are beaten by a much simpler baseline. This finding demonstrates one reason why it is important to test methods across a wide range of datasets.

Comparing the LSTM based methods to the NP it is clear that the LSTM based methods perform better in comparison to NP on larger datasets e.g. Election and Dong. Whereas the NP methods in comparison perform much better on the smaller datasets (YouTuBean), further they also perform better with respect to macro F1 scores. The macro F1 results suggest that the LSTM methods perform poorly on datasets that are highly un-balanced e.g. Mitchell and this gets worse when the dataset is small and un-balanced e.g. YouTuBean. Furthermore for the YouTuBean and the Mitchell datasets the macro F1 score distribution is quite large, as shown in figure 3.23[47], in comparison to all of the other datasets. This suggests that the LSTM methods are more sensitive to random seed/initialisation for smaller and un-balanced datasets. To overcome the overfitting to particular labels, re-sampling techniques such as under or over sampling could be of use. In some prior work it has been shown that transfer learning from a document sentiment dataset has greatly improved the macro F1 score for LSTM methods (He et al., 2018b). However more work investigating how to improve NN based TDSA methods with respect to highly un-balanced and/or low resourced corpora should be

[47]For the distribution of the accuracy scores for the LSTM methods see figure A.7 in appendix A.2.

investigated. Lastly from the significance results it is clear that the NP methods do not struggle to beat the LSTM sentence level classifier baseline.

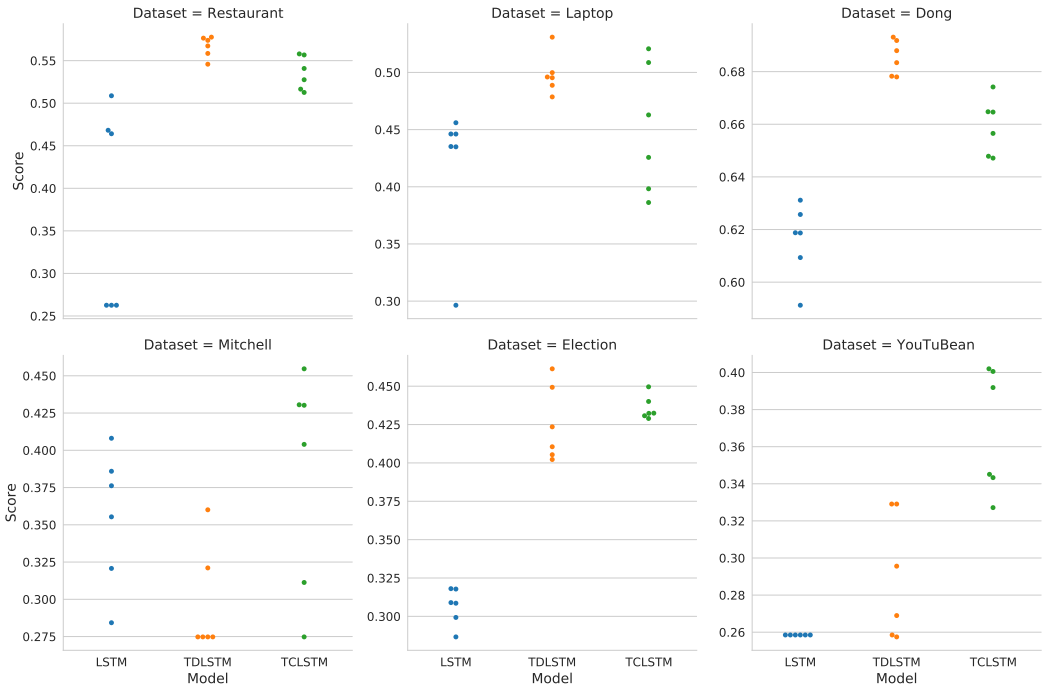


Figure 3.23: Distribution of macro F1 scores from the six runs for each LSTM method and test dataset.

From a performance perspective in general the hardest dataset, according to accuracy, is the Election one, even though it is the largest by a large margin. The reason for its difficulty is believed to stem from the fact it has a large distribution of samples in DS_2 and DS_3, as Wang et al. (2017a) suggest that DS_3 is the most difficult scenario. It also appears that the methods are less affected by the type, domain, or medium that dataset has come from but rather the size, DS distribution, and sentiment class distribution. This is observed as methods that do incorporate type specific features e.g. a Twitter based dependency parser perform worse in rank terms on the social media type datasets some of the time than on the non-social media type datasets. For example, TDParse performs worse on Dong but better on Laptop compared to TD.

It is worth noting that to a large extent the NP methods have an advantage over the LSTM methods as the C-value which is significant has been tuned for each dataset and method, where as the LSTM methods have had no tuning. However some of the NP methods still perform better than the TDLSTM results for the Restaurant dataset from prior work, as shown by table 3.12. This is highlighted as these other prior works may have tuned the TDLSTM or found a better seed value than in the evaluation performed here. To give the LSTM methods an increased advantage over the NP methods, due to the NP methods having their C-values tuned, the LSTM methods have been recompared to the NP whereby the best performing run/seed value is used. Figures 3.24 and 3.25 show the number of datasets that the methods are significantly better on using the best performing run based on the accuracy and macro F1 metric. However with this

increased advantage for the LSTM methods it can be seen that the NP methods are still significantly better and the change in the heatmap compared to figure 3.22 is minimal.

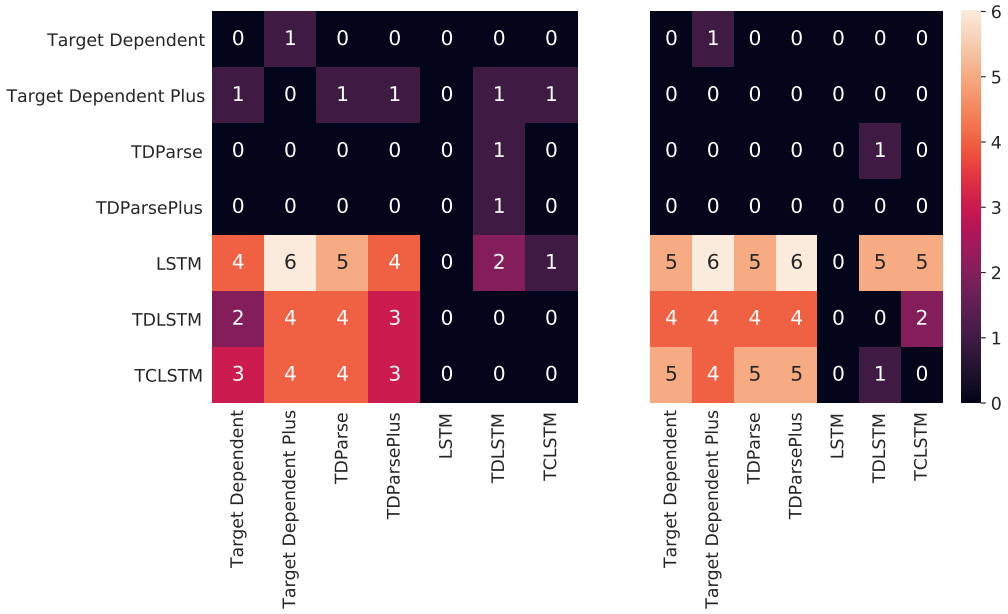


Figure 3.24: The number of datasets where the column methods are statistically significantly better than the row methods, corrected using Bonferroni. The left and right heatmaps represent the accuracy and macro F1 scores. This is using the best performing run based on the **accuracy metric** for the LSTM methods.

As stated earlier, the LSTM methods tend to perform worse on smaller datasets. Thus to test how this low resource setting affects LSTM and NP methods on a larger scale all of the datasets training set sizes have been reduced to the same size as YouTuBean training set size, which is the smallest dataset within the evaluated datasets[48]. Further for the LSTM based methods this new reduced training set size also means that 20% of that training set is used as a validation set for early stopping. All methods are retrained using the same settings. As the YouTuBean dataset has already been evaluated and would not be affected by this new size reduction it will not be included in the analysis of these experiments. The accuracy and macro F1 results on the test sets for these datasets can be seen in tables 3.18 and 3.19. The statistically significant results comparing each method can be seen for both metrics in figure 3.26, for the LSTM methods the median best performing run based on the accuracy metric is used to compare to all other methods.

From the results it is clear that the NP methods perform the best in this low resource setting. The datasets that the LSTM methods did perform better on (Dong and Election), they are now worse on. The target specific LSTM methods (TDLSTM and TCLSTM) now have very similar performance to the sentence level LSTM method, which was not the case for all of the results in the normal resource setting. Furthermore the LSTM

[48]For the Dong et al. (2014) Twitter dataset instead of training the NP methods using the median pooled approach due to the 'same target multiple appearance' issue Wang et al. (2017a). All methods including the NP methods will use the first appearance of the target in the text. This was done due to compatibility reason between the NP and LSTM methods after splitting the training dataset.

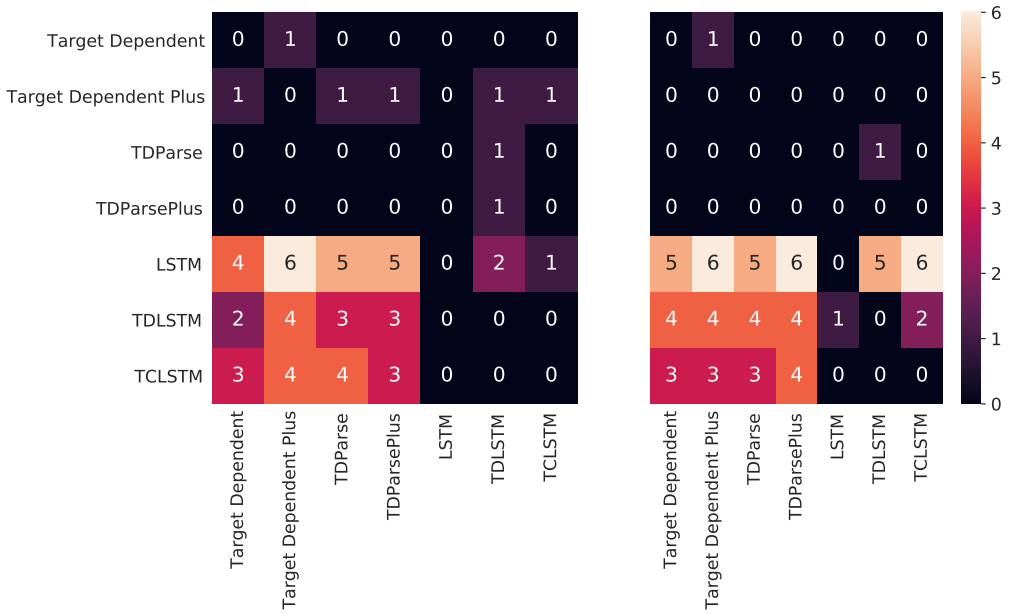


Figure 3.25: The number of datasets where the column methods are statistically significantly better than the row methods, corrected using Bonferroni. The left and right heatmaps represent the accuracy and macro F1 scores. This is using the best performing run based on the **macro F1 metric** for the LSTM methods.

Method	D	E	L	M	R	Mean
LSTM	<u>50.00</u>	<u>46.20</u>	53.11	<u>70.11</u>	<u>65.00</u>	<u>56.88</u>
TDLSTM	50.02	49.00	52.25	<u>70.11</u>	<u>65.00</u>	57.28
TCLSTM	51.01	49.91	<u>50.89</u>	<u>70.11</u>	<u>65.00</u>	57.39
TD	65.75	**51.59**	60.82	**72.75**	72.50	64.68
TD+	**66.62**	49.08	**64.58**	72.64	74.29	65.44
TDParse	65.61	**51.59**	60.82	72.44	74.11	64.91
TDParse+	65.75	50.45	64.26	71.94	**74.91**	**65.46**
Mean	59.25	49.69	58.10	71.44	70.11	-
D=Dong, E=Election, L=Laptop, M=Mitchell, R=Restaurant, Y=YouTuBean						

Table 3.18: Using the smaller training datasets, the accuracy results on the test sets of each dataset. For the LSTM based methods this is the mean accuracy result. The mean accuracy across all datasets for each method is in the right most column. Where the **bold** and <u>underlined</u> values indicate the best and worst methods for each dataset and the overall mean accuracy, respectively. The mean accuracy score for each dataset is in the last row.

methods have exceptionally poor performance on the macro F1 results, so much so that all NP methods are significantly better on all datasets compared to all of the LSTM methods. The macro F1 results are broken down into F1 scores for each of the sentiment classes positive, neutral, and negative which can be see in tables 3.20, 3.21, and 3.22.

Method	D	E	L	M	R	Mean
LSTM	<u>22.22</u>	27.15	<u>23.47</u>	<u>27.48</u>	<u>26.26</u>	<u>25.32</u>
TDLSTM	22.29	<u>26.14</u>	29.12	<u>27.48</u>	<u>26.26</u>	26.26
TCLSTM	27.65	32.46	32.59	<u>27.48</u>	<u>26.26</u>	29.29
TD	**62.48**	39.18	52.11	**45.73**	54.66	50.83
TD+	62.44	39.34	**57.71**	41.41	58.65	51.91
TDParse	62.43	**39.39**	51.69	44.14	57.77	51.08
TDParse+	61.36	38.25	56.86	44.58	**60.14**	**52.24**
Mean	45.84	34.56	43.36	36.90	44.29	-
D=Dong, E=Election, L=Laptop, M=Mitchell, R=Restaurant, Y=YouTuBean						

Table 3.19: Using the smaller training datasets, the macro F1 results on the test sets of each dataset. For the LSTM based methods this is the mean macro F1 result. The mean macro F1 across all datasets for each method is in the right most column. Where the **bold** and <u>underlined</u> values indicate the best and worst methods for each dataset and the overall mean macro F1, respectively. The mean macro F1 score for each dataset is in the last row.

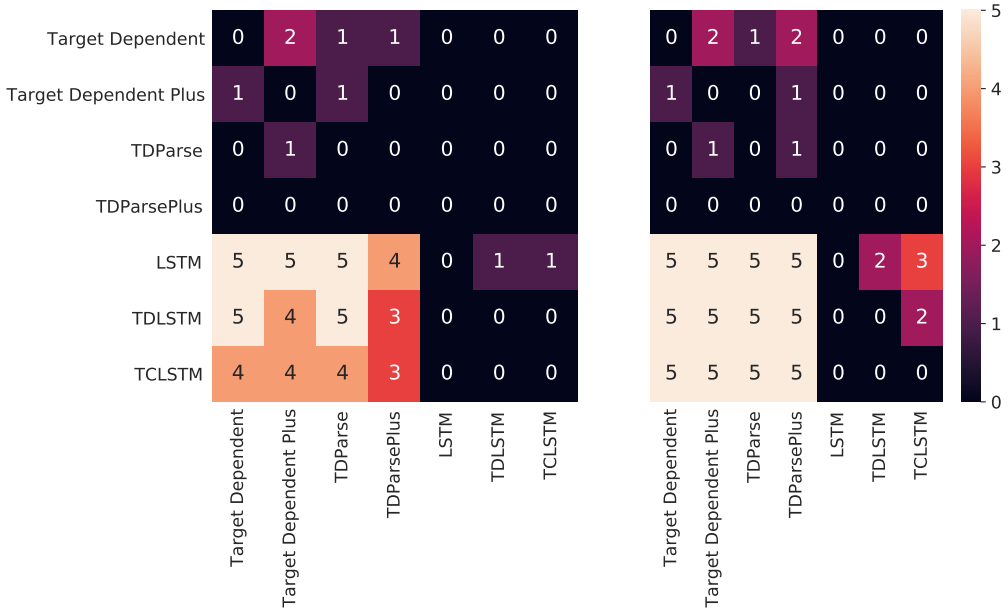


Figure 3.26: Using the smaller training datasets, the values represent number of datasets the column methods are statistically significantly better than the row methods, corrected using Bonferroni. The left and right heatmap represent the accuracy and macro F1 scores. This is using the median performing run based on the accuracy metric for the LSTM methods.

From these results it is clear that for the datasets that are most un-balanced, the LSTM methods can only predict the majority class correctly e.g neutral for Mitchell and positive for Restaurant. In comparison the NP methods are much less affected by the un-balanced

datasets and can predict at least one sample correctly for all sentiment classes for all datasets, which only the TCLSTM method can do for the Laptop and Dong dataset.

In the low resource setting for NP methods, it is still clear that the addition of a dependency parser (TDParse and TDParse+) does not make a large significance difference. In comparison it is more clear that for the dataset from the written medium and coming from the review domain (Laptop and Restaurant) that using sentiment lexicons makes a large difference for both the accuracy and macro F1 metrics. This result is most likely due to the limited data and thus the sentiment lexicon act as a good inductive bias. The reason why the lexicon based methods perform well only on these datasets is most likely due to one of the lexicons coming from that medium and type, Hu et al. (2004a) sentiment lexicon, whereas the other lexicons used are more general. This shows that in a low resource setting sentiment lexicons are useful, only if they come from the same type and medium, as the YouTuBean which comes from the review type but not written medium does not benefit from the sentiment lexicons as shown in table 3.16 and 3.17[49].

Method	D	E	L	M	R	Mean
LSTM	<u>0.00</u>	<u>0.00</u>	<u>69.43</u>	<u>0.00</u>	<u>78.79</u>	<u>29.64</u>
TDLSTM	<u>0.00</u>	<u>0.00</u>	70.53	<u>0.00</u>	<u>78.79</u>	29.86
TCLSTM	10.40	<u>0.00</u>	70.28	<u>0.00</u>	<u>78.79</u>	31.89
TD	57.88	6.38	76.73	**41.40**	84.00	**53.28**
TD+	54.23	**12.43**	78.79	31.40	84.96	52.36
TDParse	**58.20**	6.83	76.82	38.30	84.57	52.94
TDParse+	52.63	5.87	**79.31**	38.35	**85.17**	52.27

Table 3.20: Using the smaller training datasets, the F1 results for the **positive class** on the test sets of each dataset. For the LSTM based methods this is the mean F1 result. The mean F1 across all datasets for each method is in the right most column. Where the **bold** and <u>underlined</u> values indicate the best and worst methods for each dataset and the overall mean F1, respectively.

[49]For the TDParse method there is an increase for both the accuracy and macro F1 metric when adding sentiment lexicon. This is not the case for the TD method whereby adding sentiment lexicons harms the performance for both accuracy and macro F1.

Method	D	E	L	M	R	Mean
LSTM	66.67	20.71	0.00	82.43	0.00	33.96
TDLSTM	66.69	13.70	0.00	82.43	0.00	32.56
TCLSTM	67.02	34.57	0.39	82.43	0.00	36.88
TD	72.87	52.06	33.62	83.68	31.91	54.83
TD+	**73.99**	48.45	**42.80**	**83.74**	37.54	**57.31**
TDParse	72.41	**52.79**	31.62	83.38	33.21	54.68
TDParse+	73.37	50.98	41.11	83.15	**37.67**	57.26

Table 3.21: Using the smaller training datasets, the F1 results for the **neutral class** on the test sets of each dataset. For the LSTM based methods this is the mean F1 result. The mean F1 across all datasets for each method is in the right most column. Where the **bold** and underlined values indicate the best and worst methods for each dataset and the overall mean F1, respectively.

Method	D	E	L	M	R	Mean
LSTM	0.00	60.75	1.00	0.00	0.00	12.35
TDLSTM	0.19	**64.72**	16.82	0.00	0.00	16.34
TCLSTM	5.54	62.81	27.09	0.00	0.00	19.09
TD	56.70	59.11	45.97	12.12	48.05	44.39
TD+	**59.09**	57.14	**51.53**	9.09	53.46	46.06
TDParse	56.68	58.54	46.63	10.75	55.52	45.62
TDParse+	58.09	57.90	50.15	**12.24**	**57.58**	**47.19**

Table 3.22: Using the smaller training datasets, the F1 results for the **negative class** on the test sets of each dataset. For the LSTM based methods this is the mean F1 result. The mean F1 across all datasets for each method is in the right most column. Where the **bold** and underlined values indicate the best and worst methods for each dataset and the overall mean F1, respectively.

3.7 Conclusion

Within this chapter, the reproduction studies have found, for the first time, that for NP methods within TDSA both scaling features and C-values within SVMs are statistically significant factors. Furthermore it is recommended that these factors are reported within a structured format like that suggested by Dodge et al. (2019) within Appendix B-D with all other relevant information about the method. For TDSA LSTM based methods it has been shown for at least one metric (macro F1) that they can be statistically significantly affected by random seeds for the first time, this has been shown previously by Reimers et al. (2017) for neural sequence labelling methods. Thus it is recommended to follow Reimers et al. (2017) advice on reporting and comparing distribution of scores that are generated from the LSTM methods by different random seeds.

Additionally the reproduction studies found that for NP the larger general embedding (300 dimension GloVe) can perform as well as the type and/or task specific embeddings that the original method used (Vo et al., 2015). This implies that from an energy saving perspective it can be useful to train smaller more relevant embeddings for NP methods.

While the LSTM methods preferred the larger general embedding, suggesting there is no requirement to train smaller more relevant embeddings. However these findings have only been tested on NP and LSTM methods on one Twitter dataset (Dong (Dong et al., 2014)), it would be beneficial to broaden the experiment to more datasets. These findings so far in the conclusion have allowed us to answer RQ 1 'what lessons can be learned from reproducing a method within TDSA?'.

The following findings help to answer RQ 2 'how generalisable are existing methods within TDSA?'. It is found that in general the NP methods are better than the LSTM, but the LSTM methods can perform better in general on larger datasets. Thus showing to some degree that there is no one winning method. While testing methods across various datasets that have different domains, types, and mediums it is found that on the standard datasets sizes these factors do not differentiate the methods. Rather, testing methods on datasets that vary by size, sentiment class distribution, and Distinct Sentiment (DS) distribution are the most influential factors. From these factors it is shown that in general all methods perform badly when the dataset contains a large distribution of samples from DS_3 and DS_2. Further, the LSTM methods are badly affected when the datasets are highly un-balanced and/or small, in these cases the NP methods are highly recommended. It was found in the low resource setting, that the inductive bias of sentiment lexicons in NP methods is useful, only if the sentiment lexicon comes from the same type and medium. Neither, the use of sentiment lexicons nor the features from a dependency parser are of significant use in the higher resource setting. Lastly it was found in some cases the target specific methods (TDLSTM and TCLSTM) were no better than the baseline (LSTM) method. This was not found in the original paper (Tang et al., 2016b), as the datasets this was found on was not used by the original paper, demonstrating the need to test methods across varying datasets. From these findings it is recommended that future work investigates how to improve TDSA methods within the low resource and/or unbalanced setting especially for NN/LSTM based methods. He et al. (2018b) has already shown that transfer learning from document level sentiment analysis can improve LSTM based methods performance on unbalanced TDSA datasets. Thus transfer learning and or multi-task learning could be a good future direction.

There are some caveats with the research presented so far on 'how generalisable are methods within TDSA?'. Even though it was found that LSTM based methods did not perform as well as the NP methods on small or smaller datasets, this does not mean another different LSTM based method could not perform better. For instance the LSTM methods used did not have any regularisation applied[50], where regularisation methods such as variational dropout (Gal et al., 2016) and label smoothing (Szegedy et al., 2016) have been shown to enhance the performance of neural network based methods (Gal et al., 2016; Song et al., 2019). Furthermore, it should be noted that all findings here are constrained to the English language, thus the findings here are language dependent[51].

[50]This decision was made as the LSTM methods used followed the original design of the LSTM methods (Tang et al., 2016b), where the original design did not use regularisation.

[51]This is highlighted following what has been known as the #BenderRule (Bender, 2019), which re-iterated a point made in prior work (Bender, 2011), that not stating the language (normally English) the data has come from misleads the reader into thinking the work is language independent.

Chapter 4

Improving Experimental Methodology for TDSA

4.1 Introduction[1]

As clearly shown within table 3.2 of the reproducibility and generalisability chapter 3, many methods have been created, although they have only been applied to selected datasets. This lack of reporting is further compounded with many prior works only stating accuracy and/or macro F1 scores on the entire dataset. This type of reporting is good at comparing methods generally on these datasets. However having more detailed analysis on different splits of the data or task specific metrics would advance the community in knowing what the methods are not representing, for instance a method could generally perform well but might perform badly when there are lots of targets in one text. Thus the field would benefit generally from detailed error analysis. Some prior works have suggested different TDSA specific splits of the entire dataset to overcome this error analysis deficiency (Nguyen et al., 2015; Wang et al., 2017a; He et al., 2018a; Yang et al., 2018), but few publications since these have used the designed splits and only report the entire dataset score. This causes a major problem within our community, and so 'we do not know what we know'. To further expand on the meaning of this, a lot of the methods within the community have general scores that are fairly similar e.g. only showing 3-4% differences on the Restaurant dataset (table 4.1)[2] but what does that mean? It could mean that a method is better on texts that contain multiple targets with different sentiments, or the method is good at generalising to new unseen targets.

The focus of this chapter is to answer RQ 3 'What is an appropriate empirical evaluation methodology for TDSA?'. To answer this question, we first compare and contrast the different existing error analysis splits for TDSA, as well as better formalising these splits so that they can be applied to any dataset. From these existing error splits, two novel splits and a novel metric are created. These error splits are analysed through the results of three different TDSA models and one text classification model to first justify

[1] All code that creates the evidence for this chapter can be found here: `https://github.com/apmoore1/tdsa_comparisons`. Certain sections throughout this chapter may have more specific pointers to python notebooks that created the analyses within that given section.

[2] For an up to date list of papers and scores see *Papers With Code* `https://paperswithcode.com/sota/aspect-based-sentiment-analysis-on-semeval`.

to some extent what the splits are measuring, and secondly compare the different models performance on these splits and metrics as baselines. From the baseline results the error splits and metrics are reviewed and recommendations of what error splits and metrics should be used and why are given. These formalised error splits and novel metrics will allow future researchers to better understand what the models are and are not capturing without having to resort to qualitative case studies and will benefit from fair, easy, and reproducible comparisons between works. Lastly to ensure all the results and analyses are performed on varying and standard datasets in the field, all experiments in this chapter are performed on three public English datasets: 1. SemEval 2014 Laptop (Pontiki et al., 2014) (Laptop), 2. SemEval 2014 Restaurant (Pontiki et al., 2014) (Restaurant), and 3. Election Twitter (Wang et al., 2017a) (Election). All three of these datasets have been introduced in chapter 3 section 3.4, and overall dataset descriptions and statistics can be found in tables 3.3 and 3.4.

Model	Laptop		Restaurant	
	Accuracy	macro F1	Accuracy	macro F1
ATAE-LSTM (Wang et al., 2016b)	69.27	-	78.50	-
TDLSTM (Tang et al., 2016b)	<u>68.83</u>	<u>68.43</u>	<u>78.00</u>	<u>66.73</u>
MemNet (Tang et al., 2016a)	72.37	-	80.32	-
IAN (Ma et al., 2017)	72.10	-	78.60	-
RAM (Chen et al., 2017)	75.01	70.51	79.79	68.86
MGAN (Fan et al., 2018)	75.39	72.47	81.25	71.94
TNet (Li et al., 2018d)	76.54	71.75	80.79	71.27
PBAN (Gu et al., 2018)	74.12	-	81.16	-
Cabasc (Liu et al., 2018)	75.07	-	80.89	-
IACapsNet (Du et al., 2019)	**76.80**	**73.29**	**81.79**	**73.40**

Table 4.1: Previous results for models that only use GloVe word embeddings. The **bold** and <u>underlined</u> values represent the best and worse performing score for each metric and dataset. This table has been taken from Du et al. (2019).

4.2 Error Analysis Background

As stated in the introduction of this chapter, TDSA error analysis has been lacking in both the reporting and the error splits available to analyse the methods. Therefore in this section the thesis will review the current error analysis splits available, as well as creating new error splits, and lastly stating some hypotheses around what these splits actually mean and therefore why they are useful (see table 4.2 for a full summary). The hypothesis and use around the splits will then be tested in the baseline experiments subsection 4.3.3.

4.2.1 Previous Work

Currently there have only been four unique error splits suggested for TDSA and these splits can be grouped into two substantially different error split groups. The first suggested split was by Nguyen et al. (2015) which is based on the number of targets per text, this

split created three different subsets of data: *ST1* contains samples that only have one target per text, *ST2* and *ST3* include samples that have more than one target per text but *ST2* is restricted to samples that contain one sentiment within the whole text. The second split named Distinct Sentiment (DS)[3] by Wang et al. (2017a) which is very similar to the first, is based around the number of unique sentiments per text: DS_1, DS_2, and DS_3 would be all the samples that contain only one, two, and three sentiments in the text, respectively. The third split which is denoted as NT in this thesis has been used in a couple of prior works (He et al., 2018a; Zhang et al., 2019a), this split divides the data by the number of targets per text. Each of the two prior works use different subsets, Zhang et al. (2019a) suggests to only subset on sentences containing up to seven targets[4], and He et al. (2018a) subsetted by sentences containing 1, 2, 3, and more than 3 targets. From the two different NT split prior works the work by Zhang et al. (2019a) will be compared to the most in this thesis as it is the work that contains the most subsets (7), and thus more fine grained results. These three splits are similar as they are all based around the number of targets and for the former two splits their sentiment class within a text. Furthermore, the combination of the DS_2 and DS_3 subsets is equal to the *ST3* subset. The DS_1 subset is equal to the combination of *ST1* and *ST2*. Also when there is only one target in the text NT_1 this is equal to *ST1*. Lastly if you create subsets by conditioning on NT_i and DS_1 where $i > 1$ this would be equal to *ST2*. Following on from these works, Xue et al. (2018) created the *Hard* subset which is in fact the same as the *ST3* subset[5] and therefore in this thesis is not counted as a new split as it is a direct derivative of past splits. All of these splits are relatively local splits in the sense that they do not take into account the global information of what is in the entire training or test datasets. Furthermore, all of these splits have been created to analyse the difficulty of a sample based on the number of targets and sentiments in the text, where it has been shown at least that more unique sentiments causes samples to be more difficult (Wang et al., 2017a; Nguyen et al., 2015) but the same cannot be said about more targets (Zhang et al., 2019a; Nguyen et al., 2015). From one of the original papers that introduced NT (Zhang et al., 2019a), where the main objective of this split is to increase the number of targets explicitly and the number of unique sentiments is not taken into account, the difficulty of the samples does not increase when the number of targets increase (see figure 4 in Zhang et al. (2019a)). Lastly even through the DS_i split was stated to get more difficult as i increased and has been shown in original work to be true (Wang et al., 2017a), it has also been shown in the same work not to be true when either the method and or metric changes (see table 4 in Wang et al. (2017a)). Both of these unexpected empirical findings for the NT and DS split will be explored empirically in the baseline results subsection 4.3.3.

The ST, DS, and NT splits are the first group of the two substantially different group splits, the second group of splits only contains one split by Yang et al. (2018), which in this thesis is called the *n-shot* split. Yang et al. (2018) were the first to explore subsets of data based on the number of times the target/aspect has appeared in the training data compared to the test. However it is not the first time in NLP that this type of error analysis has been done as this is just an adaptation of the *n-shot* learning setup

[3]Distinct Sentiment has already been introduced and used in chapter 3, section 3.4.

[4]That is seven subsets where each subset contains; 1, 2, 3, 4, 5, 6, or 7 targets per sentence.

[5]And as stated earlier in this paragraph the same as the combination of the DS_2 and DS_3 subsets

which has been performed in text classification (Zhang et al., 2019a), multi-lingual target extraction (Jebbara et al., 2019), entities and relationships within Natural Language Inference (NLI) (Levy et al., 2017), and finally many other NLP tasks while evaluating a language model (Radford et al., 2019). This *n-shot* setup takes the test and training datasets and finds the number of times (n) the target in the test dataset appears in the training. Therefore the subset of *zero shot* ($n = 0$) targets would be all the targets in the test that never appear in the training dataset. The *n-shot* setup allows the method to be tested for its ability to generalise to unseen targets, and its capability of learning a new target. Thus a model that can generalise well should be able to perform well with few (or no) target examples in the training data. Furthermore, the expectation would be as n increased the samples within that n would get easier to classify, as it has been shown in other areas of NLP that *zero-shot* ($n = 0$) contains the most difficult samples (Jebbara et al., 2019). However, findings of the original work by Yang et al. (2018) found that, in general, model performance does not correlate with n, thus indicating that *zero-shot* is no more difficult than any other n to classify. This finding is unexpected and will be tested further in subsection 4.3.3.

As suggested earlier, the main difference between these two split groups is that the first uses local information whereas the latter uses the global information between the test and training datasets. In the next subsection, the new data splits will be created to complement these existing splits. Given that the first split group contains three slightly different splits, in this thesis only the *DS* and *NT* splits will be used. These two were chosen as they complement each other well as the *NT* split measures the effectiveness of a method with respect to the number of targets in a text and therefore their interactions. The *DS* split measures a method's ability to identify target sentiment relations. Thus if a method performs well on the *NT* subsets but not the *DS* it suggests that it can understand when targets should interact with each other such as in the conjunction case e.g. 'The *battery* is really good and so is the *screen*'[6], but it is not good at identifying target sentiment relationships. Thus the *ST* split is not required as it measures to some extent both *DS* and *NT*.

4.2.2 New splits

In this thesis, two new splits are suggested, one based on global information the other local information. The global split denoted as Target Sentiment Relation (*TSR*) focuses on different ways to probe a method's ability to generalise to new targets and to new sentiment relations for already known targets. The local split denoted as Target Sentence Sentiment Ratio (*TSSR*) measures the combination of *DS* and *NT* splits but taking into account the number of different sentiments rather the unique sentiments within the *DS* split. Therefore it can be used to measure the affect to some extent of overfitting to the most frequent sentiment within a sentence.

The *TSR* split contains three subsets: 1. Known Sentiment Known Target (*KSKT*), 2. Unknown Sentiment Known Target (*USKT*), and 3. Unknown Targets (*UT*). The first subset should be the easiest as the method will have at least some information on both the target and how it relates to that sentiment label, thus this sets the threshold for the other two subsets to meet. The second is potentially the most difficult for the

[6] 'battery' and 'screen' are the two targets.

method as it would have seen the target and a relation to another sentiment label, but not this specific label. Therefore the *USKT* split tests how well a method can generalise to new sentiment relations without overfitting to known relations for that target. The last subset is the same as the *zero shot* case in the *n-shot* split, thus it tests the ability of the method to generalise to new targets. This split can be seen to relate to the relation extraction task of performing zero shot entity extraction for the *UT* subset and zero shot relation extraction for the *USKT* subset (Levy et al., 2017). The results from the relation extraction literature (Levy et al., 2017; Abdou et al., 2019) motivates the reason why *USKT* should be the most difficult split and the *UT* to be easier.

The *TSSR* is based on each target's sentiment frequency within a sentence thus taking into account both unique sentiments and number of targets within a sentence. Before stating how *TSSR* is calculated some notation is required; given a target that is represented as t_{ji} where j denotes the sentence index from all of the sentences $X = \{x_1, ..., x_k\}$, and i denotes the index of the target within sentence j where the target is 1 of n targets $T_j = \{t_{j1}, ..., t_{jn}\}$ within sentence j. Furthermore the sentiment value for t_{ji} comes from the set $S = \{s_1, ..., s_i\}$ of all sentiment values, where the sentiment value for a target comes from the following function $Sent(t_{ji}) \subset S$. Given this notation *TSSR* can be calculated from equation 4.1[7], thus the subsets within *TSSR* are a continuous value ranging from 0 to 1. Where the minimum *TSSR* value would come from one of two types of sentences:

1. A sentence that contains lots of targets but few or one come from only one unique sentiment class.

2. A sentence that contains few targets but all targets come from a different sentiment class.

$$TSSR(t_{ji}) = \frac{\sum_{n=1}^{|T_j|}[Sent(t_{ji}) = Sent(t_{jn})]}{|T_j|} \tag{4.1}$$

Targets that are within the 1 subset have to be either the only target within a sentence (NT_1) or a sentence that contains multiple target all with the same sentiment, thus the 1 subset is equal to the combination of *ST1* and *ST2* subsets or the DS_1 subset. Any *TSSR* subset that is less than 1 must come from targets that are within texts that contain more than one unique sentiment and thus more than one target hence part of either DS_2 or DS_3 subsets. Furthermore, the assumption is that the lower the subset value the more difficult the sample will be to classify, as the target must have come from a sentence that contains lots of targets and/or the target itself has a very rare sentiment value within that sentence. For instance, a sentence that contains 3 targets and 3 different sentiments will all be part of *TSSR* subset $\frac{1}{3}$. Another example, a sentence that contain 3 targets where the first is positive and the rest negative, the first will be part of *TSSR* subset $\frac{1}{3}$ where as the other two parts subset $\frac{2}{3}$. TSSR's main purpose is to detect overfitting to the most frequent sentiment within the sentence, of which this can be measured by taking the absolute difference in some metric, e.g. accuracy, between high and low TSSR values. Whereby, a large absolute difference would suggest overfitting whereas a low

[7] A *TSSR* value is assigned to each target and is calculated per target.

would not, as performing well on the high TSSR values could come from just predicting the most frequent sentiment in the sentence.

The *TSSR* split can be seen to be very similar to the *DS* split as they both measure to some degree the number of sentiments within a sentence. However the main point of *TSSR* unlike *DS* is to measure overfitting to the most frequent sentiment within the sentence, where as *DS* better evaluates the target sentiment relationship as the subset explicitly measure this. The *TSSR* and *DS* splits can be used together to measure the target sentiment relation better, as when the *TSSR* shows overfitting this informs how reliable the metrics within the *DS* subsets are. For example, when overfitting through *TSSR* is high, the values of the *DS* split may be unreliable as the model is not learning the target sentiment relationship, but just how well it is at overfitting.

Summaries of the differences in the error splits introduced in this section and section 4.2.1 can be seen in table 4.2. Examples for each of these splits and subsets can be found in table 4.3.

Split	Description	What it measures
Distinct Sentiment (*DS*)(Wang et al., 2017a).	Splits the data based on the number of unique sentiment classes within one text where DS_i represents i unique sentiment classes within a text.	The method's ability to capture sentiment relations within the text. If the method performs well on larger values of i the better the method is at capturing these relations. Less directly this measures the method's capability of modelling target interactions.
NT (Zhang et al., 2019a).	Splits the data based on the number of targets within the text, where NT_i split contains all the texts that have i number of targets within it.	The method's ability to model target interactions, where the larger i is the higher the likelihood of more target interactions. For instance, in example 4 from table 4.3 to infer the sentiment for *Dave* you need to know the sentiment towards *police* and *crime*.
TSSR novel split.	Splits the data based on the *TSSR* equation 4.1. The maximum value of 1 represents a target that is within a text that contains one unique sentiment. A *TSSR* value less than 1 denotes a target that is within a text that contains at least more than one unique sentiment. A *TSSR* value gets smaller based on how unique the targets sentiment is within that text.	Capturing overfitting to the most frequent sentiment within a text, which can be measured to some degree by comparing the method's High (Multi 1) *TSSR* subset to the Low (1) *TSSR* subset. It can also to some degree measure both target interaction and sentiment relations when the *TSSR* value is less than 1, thus a combination of both *DS* and *NT*.
ST (Nguyen et al., 2015).	Splits the data into three subsets *ST1* for texts that contain one target, *ST2* and *ST3* for texts that contain more than one target, where *ST2* texts only have one unique sentiment.	The method's ability to capture sentiment relations and its interaction with other targets.
n-shot (Yang et al., 2018).	Splits the data based on the number of times the target has appeared in the training data, when $n = i$ the subset contains samples that have targets that only appear in the training data i times.	A method's ability to generalise to unseen targets when $n = 0$, as well as its capability of learning a new target.

TRS novel split.	Splits the data into three subsets: 1. *Unknown Targets*(UT) when the target has never been seen in the training data, 2. *Unknown Sentiment Known Target*($USKT$) when the target has been seen in the training data but not with the same sentiment class, and 3. *Known Sentiment Known Target*($KSKT$) when the target has been seen in the training data with the same sentiment class. NOTE UT is equal to *n-shot* when $n = 0$.	A method's ability to generalise to unseen targets UT split, and unknown sentiment relations $USKT$ split, where the $KSKT$ split can be seen as the method's upper limit for the former two subset's performance.

Table 4.2: Summary of the different error splits.

Sample	Error subsets for each target
Examples in training data	
Example 1	
Conservatives$_1$ cut the police$_2$ budget and this cut crime$_3$! Maybe not... #spin #BattleForNumber10	All targets - DS_2, *ST3*, NT_3. Target 1 and 2 - *TSSR* value of $\frac{2}{3}$. Target 3 - *TSSR* value $\frac{1}{3}$.
Example 2	
Absolutely - taxation$_1$ is the problem #bbcqt	All targets - DS_1, *ST1*, NT_1, *TSSR* value 1.
Example 3	
Let the debate begin need a leader with tolerance for immigrants$_1$/lightbluerefugees$_2$ #BattleForNumber10	All targets - DS_1, *ST2*, NT_2, *TSSR* value 1.
Examples in test data	
Example 4	
20% less police$_1$ equals 20% less crime$_2$...wow Dave$_3$'s a genius... #BattleForNumber10	All targets - DS_1, *ST1*, NT_3, *TSSR* value 1. Target 1 - 1-shot, *USKT*, Target 2 - 1-shot, *KSKT*, Target 3 - 0-shot, *UT*.
Blue = positive, Grey = neutral, and Red = negative sentiment.	

Table 4.3: Examples of TDSA samples split into training and test datasets, where each example states the error split that the target will be put within. All examples have come from the Election Twitter dataset (Wang et al., 2017a).

4.2.3 Analysing the splits[8]

Given these five different splits *DS*, *NT*, *TSSR*, *n-shot*, and *TSR*, the analyses of how often they occur within the three main English datasets that are being examined in this chapter will be explored, through this exploration it will uncover in more detail how these datasets differ. All analysis shown in this subsection is performed on the test set of the datasets, this was chosen over a validation set as no standard/formal validation sets have been created for these datasets[9].

[8]All empirical tables and graphs within this section have been generated through the following notebook `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/TDSA_Error_Analysis.ipynb`.

[9]It has also been shown that using a standard training and test split is not optimal if the aim is to find generalisable results (Gorman et al., 2019; Moss et al., 2019). Thus, showing analysis for a validation split is somewhat pointless as the best way of evaluation in the future would be to run models across

The DS split subsets can be seen in both table 4.4 and figure 4.1 which clearly shows a massive difference between the Election dataset and the Restaurant and Laptop datasets. The Election dataset contains almost a 50-50 split between the DS_1 and DS_2 subsets compared to the almost 80-20 split within the Restaurant and Laptop datasets. This first shows that the Election dataset is likely to be a much more difficult dataset due to the model having to properly reflect the sentiment relation between targets and their corresponding sentiment within the text. This is in contrast to the samples that come from the DS_1 subset which could potentially be classified by a sentence level sentiment classifier as all the targets within that sentence all contain the same sentiment. The number of DS_1 samples within the Restaurant and Laptop dataset could explain why sentence level classifiers have performed well in TDSA tasks, which has been shown by numerous studies (Tang et al., 2016a; Wang et al., 2016b; He et al., 2018a; Jiang et al., 2019). Lastly, this also shows that the DS_3 subset, the most difficult, should not be analysed on the Laptop and Restaurant dataset due to the low number of samples.

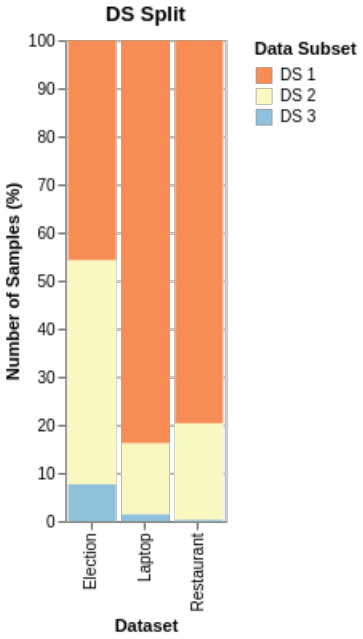


Figure 4.1: Percentage of samples per DS data subset.

	Data Subset		
Dataset	DS 1	DS 2	DS 3
Election	1164	1182	195
Laptop	535	94	9
Restaurant	892	225	3

Table 4.4: Number of samples within each DS data subset.

The NT split, unlike the DS split, has a continuous number of subsets as i of NT_i is determined by the dataset. Figure 4.2 shows unsurprisingly that the Election dataset compared to the other two contains far fewer samples that only have one or two targets per text, which most likely relates to the fact that it has far fewer DS_1 samples. Thus, showing again that the Election dataset is likely to be a more difficult dataset, as a greater number of the samples will contain sentiment that are linked through target interaction. However, as can be seen in figure 4.2 and as stated in the work suggesting this split

multiple splits and random seeds (Moss et al., 2019) when computational costs lower. For now, our evaluations only take into account random seeds subsection (4.3.2).

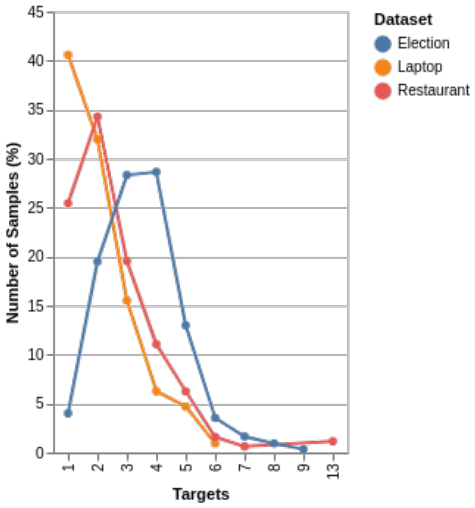


Figure 4.2: Percentage of samples per NT_i subset.

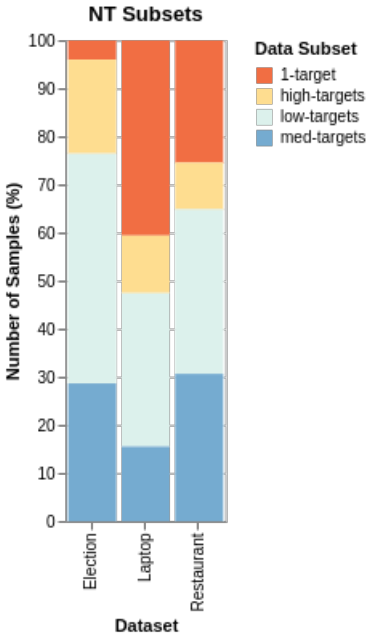


Figure 4.3: Percentage of samples per NT data subset.

(Zhang et al., 2019a) some of the subsets contain very few samples, thus any analysis on these subsets can be fairly unstable. Furthermore, as the dataset contains different ranges of i in NT_i, binning the NT split into four subsets will allow the datasets to be more comparable. The four subsets suggested would first be comprised of one subset where $i = 1$ denoted. Then the other three subsets would be made up of low, medium, and high number of i, where i values for each of these subsets are based on the equal amount of samples per i. Each of these subsets will be termed *1-target*, *low-targets*, *med-targets*, and *high-targets* respectively. These subsets can be seen in figure 4.3 and table 4.5, along with the values of i that each subset represents in table 4.6. The *1-target* subset denotes the methods baseline performance when not having to consider target interaction, which to some extent should set the upper bound for all other subsets. The other three subsets represent low to high likelihood of requiring target interaction knowledge to classify the samples correctly. Thus, a method that can model these interactions well should perform well across all four subsets, whereas a method that cannot will perform increasingly better from high i to $i = 1$.

	Data Subset			
Dataset	1-target	low-targets	med-targets	high-targets
Election	102	1216	728	495
Laptop	259	204	99	76
Restaurant	285	384	343	108

Table 4.5: Number of samples per NT subset.

The *TSSR* split has a continuous number of subsets based on the dataset, of which

	Data Subset			
Dataset	1-target	low-targets	med-targets	high-targets
Election	1, 1	2, 3	4, 4	5, 9
Laptop	1, 1	2, 2	3, 3	4, 6
Restaurant	1, 1	2, 2	3, 4	5, 13

Table 4.6: Range of i values that represent each NT subset.

the $TSSR$ subset values can range from 1 to 0. These $TSSR$ subsets can be seen in full in figure 4.4 for each dataset. As noted before in section 4.2.2 the 1 $TSSR$ subset is equal to the DS_1 subset of which this subset dominates the Laptop and Restaurant datasets. Thus for $TSSR$ value subsets less than 1 the Laptop, Restaurant, and Election datasets contain $\sim 20\%$, $\sim 24\%$, and $\sim 55\%$ of their samples respectively. Furthermore, it can be seen clearly from figure 4.5 and 4.4 that only the Election dataset has a large number of samples that have a $TSSR$ value less than or equal to 0.5. These lower $TSSR$ values (≤ 0.5) represent the least or joint equal frequent sentiment within the text when the text comes from the DS_2 subset of which the majority of samples that contain more than one unique sentiment do come from the DS_2 for all datasets. Furthermore the lower $TSSR$ value subsets are going to be by far the more difficult samples to predict for, due to them containing targets that meet at least one of the two following criteria:

1. The target coming from a text that contains lots of other targets but the target itself is within the least dominating sentiment class within the text.

2. The target comes from a text that contains few targets but all targets come from a different sentiment class.

Thus these two criteria therefore will measure the method's ability to identify target sentiment relations in a text that is dominated by other target sentiments.

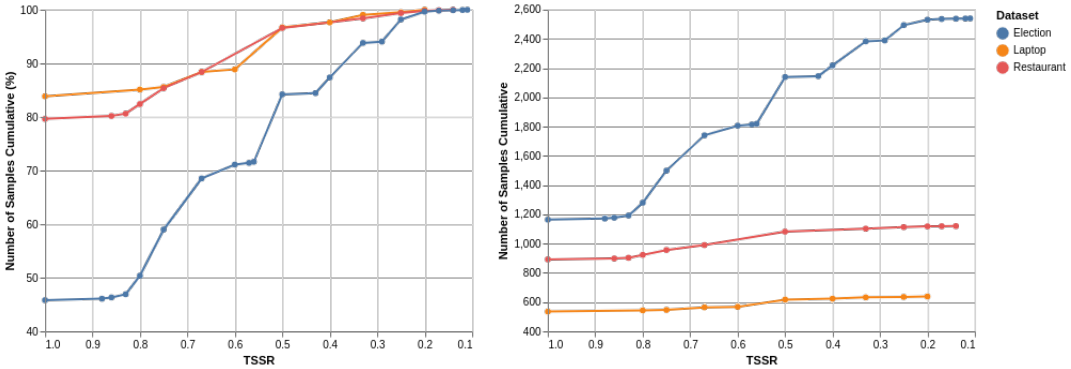


Figure 4.4: The right (left) plot shows the cumulative sample count (percentage) for decreasing values of $TSSR$.

As can be seen from the figures of 4.5 and 4.4 the $TSSR$ value subsets are different depending on the dataset just like the NT split. Therefore, the split will contain four subsets to represent different levels of difficulty and allow the performance on the subsets to be comparable across datasets. The four subsets are the following:

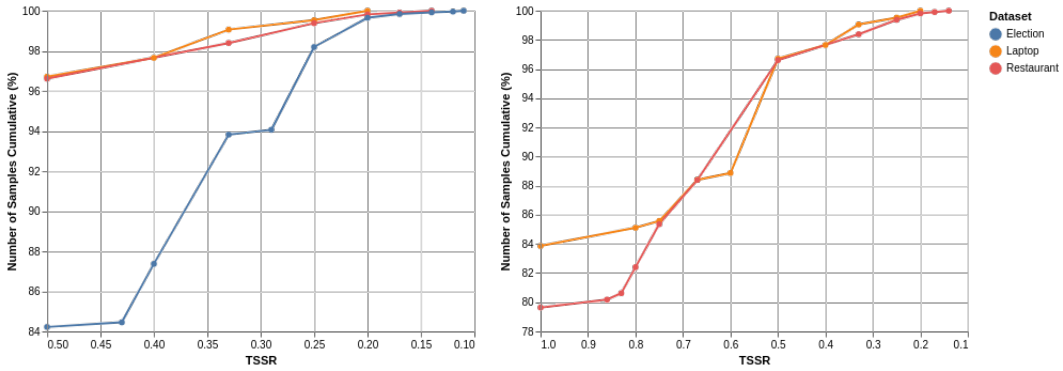


Figure 4.5: The left plot shows the cumulative sample count percentage for decreasing values of *TSSR* starting from 0.5. The right plot shows the same as the left but for the full range of *TSSR* values but excluding the Election dataset.

1. *1-TSSR* contains all targets that come from a sentence that only contain that one target thus this is the same as the NT_1/NT *1-target* subset and *ST1* subset from section 4.2.1.

2. *1-multi-TSSR* contains all targets that come from a sentence that contains more than one target but they all have the same sentiment value. This is the same as the difference of the DS_1 subset and the NT_1/NT *1-target* subset, and the same as the *ST2* subset from section 4.2.1.

3. *high-TSSR* contains all of the targets that are within the following *TSSR* value range $m \leq x < 1$.

4. *low-TSSR* contains all of the targets that are within the following *TSSR* value range $0 \leq x < m$.

The value for m is dataset specific and is based on ensuring that the *low-TSSR* values contain at least 50% of the samples after the *1-TSSR* and *1-multi-TSSR* samples have been removed from the dataset. The rest of the samples are given to the *high-TSSR* subset.

The *1-TSSR* subset explores the effect of having no target interaction nor complex target sentiment relation due to only having one target and one sentiment. The *1-multi-TSSR* explores the effect of having target interaction with potentially easier target sentiment relation due to the targets having all of the same sentiment, thus allowing the method to get the target sentiment relation incorrect without any consequence. The *high-TSSR* and *low-TSSR* measures the effect of increasing the target interaction and target sentiment relation to a smaller and larger degree respectively.

The *1-multi-TSSR* is expected to be the easiest subset as a method can make use of the target interaction and incorrectly inferring the correct target sentiment relationships as all targets will have the same sentiment. The next easiest subset would be the *1-TSSR* due to no target interaction nor complex target sentiment relationships, however more difficult than *1-multi-TSSR* as the method will have less sentiment signal within the text. Lastly the *high-TSSR* and then *low-TSSR* should be the most difficult subsets in that

order due to target interactions and complex target-sentiment relationships. Furthermore, as stated earlier in section 4.2.2 the main purpose of this split is to measure overfitting to most frequent sentiment within the text. Through the subsets the overfitting could be detected if the method performs better in the *1-multi-TSSR* than the *1-TSSR*, this could show that either the method is exploiting the frequent sentiment situation or it is better at capturing the target interaction. However, the difference in performance between the *high-TSSR* and *low-TSSR* could detect this overfitting better due to the *low-TSSR* coming from the most infrequent sentiment within the texts and the *high-TSSR* coming from the most frequent. Thus if the difference between the *high-TSSR* and the *low-TSSR* is large then most frequent sentiment overfitting is more than likely occurring.

Within figure 4.6 and table 4.7 is the breakdown of the number of samples per *TSSR* subset, table 4.8 shows the *TSSR* value range for each subset[10] (the gap in range between each subset exists because there are no samples that contain a *TSSR* value in that gap). As shown in the figure and tables, the Election dataset compared to the other two contains far fewer *1-TSSR* samples compared to the number of samples in *1-multi-TSSR*. Furthermore, the Election dataset also contains a large *high-TSSR* subset, and low compared to the other datasets.

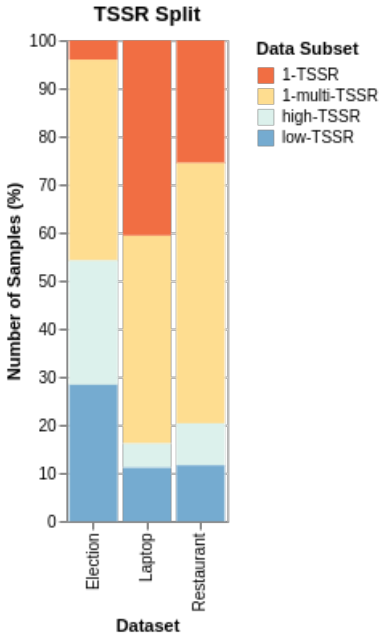


Figure 4.6: Percentage of samples per *TSSR* subset.

To further understand the relationship between the *TSSR* split and the *NT* and *DS* splits figure 4.7 shows the breakdown of each *TSSR* subset by the *NT* and *DS* splits. As can be seen from the figure for the *1-multi-TSSR* subset, the Election dataset has a more even distribution of samples that come from sentences that contain 2, 3, and 4 targets. Compared to the Restaurant and Laptop datasets where the majority of samples contain only 2 targets and then dramatically decreases as *NT* increases. For all datasets the *high-TSSR* subset contains relatively more samples that come from texts that contain a

[10] *1-multi-TSSR* is not within this table as it has the same range as *1-TSSR*.

	Data Subset			
Dataset	1-TSSR	1-multi-TSSR	high-TSSR	low-TSSR
Election	102	1062	656	721
Laptop	259	276	32	71
Restaurant	285	607	98	130

Table 4.7: Number of samples within each *TSSR* subset.

	Data Subset		
Dataset	1-TSSR	high-TSSR	low-TSSR
Election	1, 1	0.88, 0.56	0.5, 0.11
Laptop	1, 1	0.8, 0.6	0.5, 0.2
Restaurant	1, 1	0.86, 0.67	0.5, 0.14

Table 4.8: Range of *TSSR* values for each *TSSR* subset.

larger number of targets compared to the *low-TSSR* subset. This is most likely due to the fact that targets with rare sentiment within the target's sentence only occur once or twice in a large *NT* sentence where as the lesser rare sentiment targets occur far more often in those sentences and are counted within the *high-TSSR* subset. An example of this can be thought of where the sentence contains 5 targets of which 1 comes from the positive sentiment class and the rest negative, thus the *high-TSSR* will have 4 samples where as the *low-TSSR* only 1 from the sentence coming from the 5 *NT* subset. Unsurprisingly the majority of the limited number of DS_3 samples are almost all inclusively within the *low-TSSR* subset for all datasets. Figure 4.7 also shows that the main difference between the three subsets is less about the distribution of *NT* but rather the distribution of sentiment labels within a sentence. Lastly, figure 4.7 explains to some degree the reason why Election and Restaurant datasets have a large *TSSR* value range as shown in table 4.8, as these datasets must contain sentences that have a large number of targets as well as those sentences containing a different number of sentiments.

Similar to *NT*, and *TSSR* the *n-shot* split has different values of n based on the dataset, of which this can be best seen in figure 4.8. From the left plot in figure 4.8 the sharpness of each curve for each dataset would appear to relate to the size of the dataset, as Laptop is the smallest and has the sharpest curve, whereas Election is the largest and has the least steep curve. Furthermore, in the left plot it can be seen that both Restaurant and Election start to flatten off around 80% and 64% suggesting that the test dataset contains a lot of samples with targets that have been very frequently seen in the training dataset. Thus these high n samples should be easy to classify due to the method having seen lots of samples of those targets in the training dataset. To better capture how many samples with no target in training data appear compared to those with targets that appear very often, figure 4.10 contains three plots showing the low, medium, and high frequencies of n (note that the values on the Y-axis are different). From this we can clearly see that for all datasets the largest number of samples occurs when $n = 0$. Lastly, it shows that both the Restaurant and Election datasets contain a lot of samples where the targets have been seen very frequently in the training dataset.

This analysis suggests that the *n-shot* split should be broken up into four different

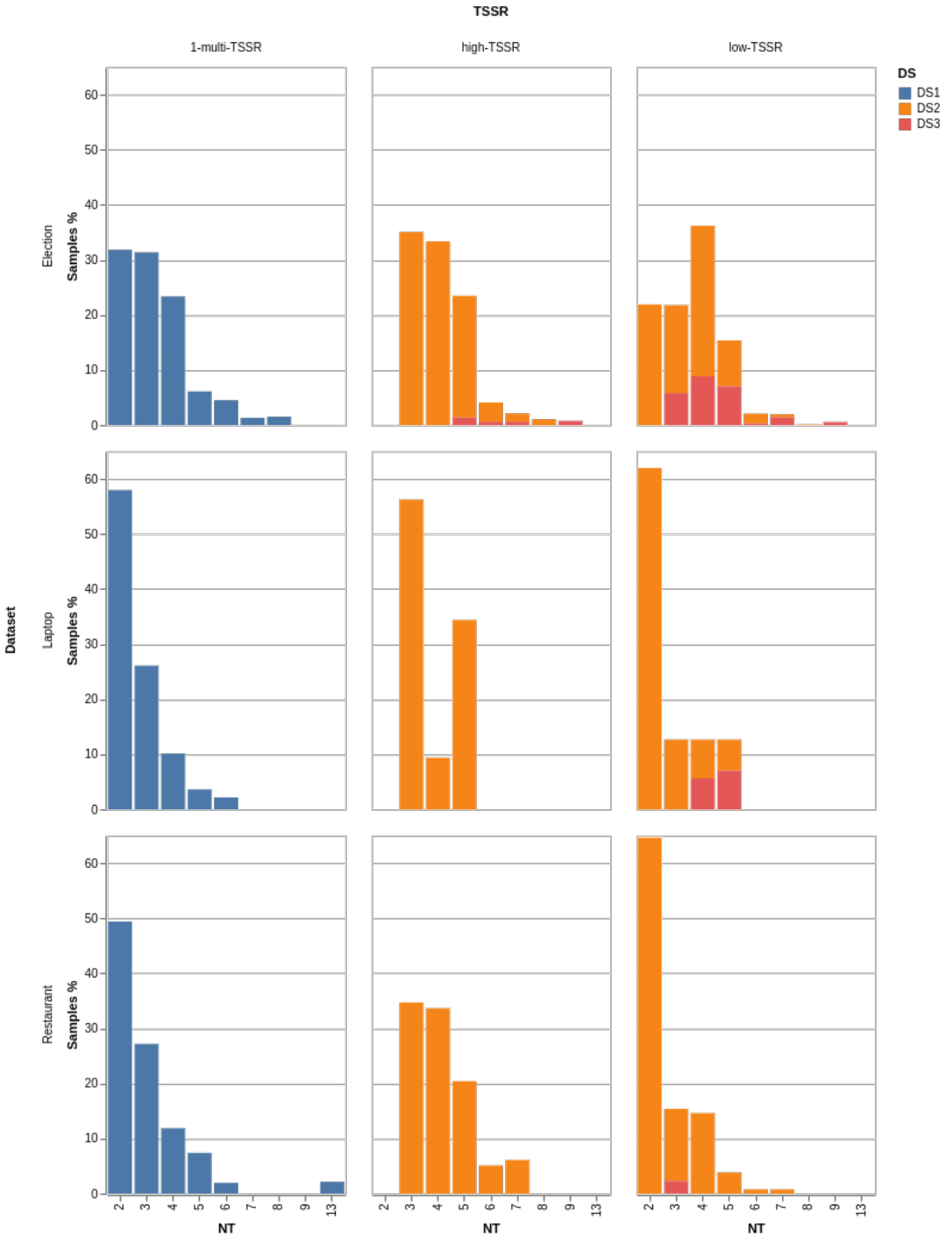


Figure 4.7: Percentage of samples per *TSSR* subset broken down by *NT* and *DS* splits.

subsets rather than n subsets, similar to the *NT* split. This is suggested as a comparison of n subsets with respect to some metric can be very unstable as some of these n subsets can contain very few samples e.g. when $n = 12$ for the Laptop dataset (see figure 4.10). The four subsets suggested would first comprise of one subset where $n = 0$ which is

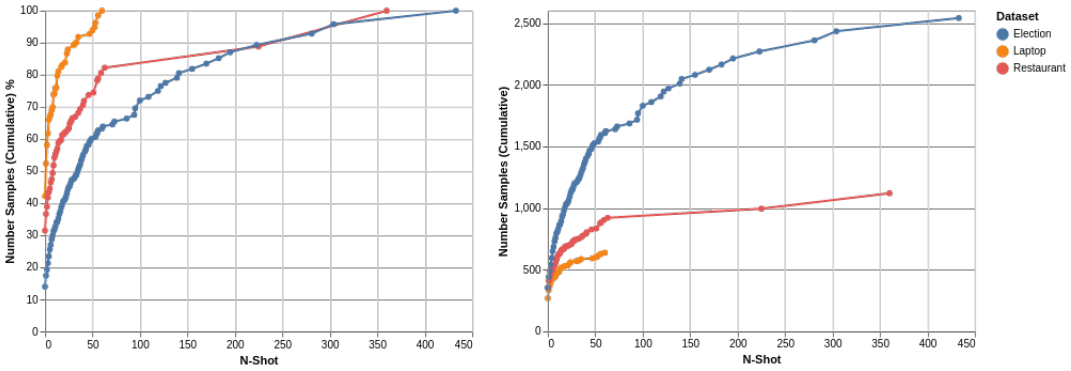


Figure 4.8: The right (left) plot shows the cumulative sample count (percentage) for increasing values of n.

the *UT* and *zero-shot* case. Then the other three subsets would be made up of low, medium, and high number of n, where n values for each of these subsets are based on the equal amount of samples per n. Each of these subsets will be termed *zero-shot*, *low-shot*, *med-shot*, and *high-shot* respectively. These subsets can be seen in figure 4.9 and table 4.9, along with the values of n that each subset represents in table 4.10. Lastly these subsets should allow for comparability across datasets and better analysis of the method's ability of learning a new target, as a good method should have a steady high performance across all of these subsets, whereas a method that overfits to targets would have a decreasing performance from the high to zero shot subsets.

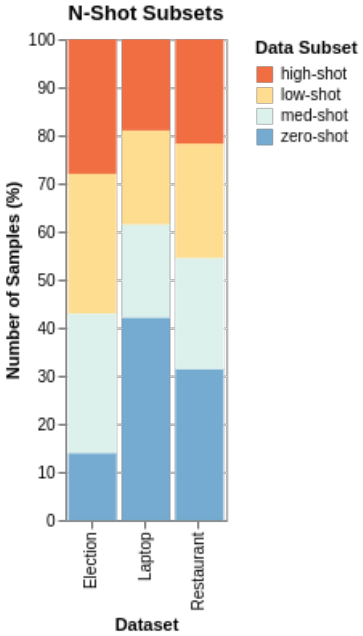


Figure 4.9: Percentage of samples per *n-shot* data subset.

TRS, the last and new split to be analysed can be best seen through figure 4.11 and table 4.11. As can be seen the *UT* subset is almost the largest subset for the smallest

	Data Subset			
Dataset	zero-shot	low-shot	med-shot	high-shot
Election	354	740	736	711
Laptop	269	125	123	121
Restaurant	352	267	258	243

Table 4.9: Number of samples per *n-shot* subset.

	Data Subset			
Dataset	zero-shot	low-shot	med-shot	high-shot
Election	0, 0	1, 23	24, 100	109, 433
Laptop	0, 0	1, 3	4, 14	17, 60
Restaurant	0, 0	1, 11	12, 55	56, 360

Table 4.10: Range of n values that represent each *n-shot* subset.

dataset (Laptop) but a relatively small subset for the largest dataset, of which this has already been seen through the *zero-shot* subset. Furthermore, the *USKT* is the smallest subset by far across the datasets, however it still counts for at least 10% of all data in the Laptop dataset. Based on the two global splits, *TRS* and *n-shot*, they show the importance of a method to generalise to new targets (*UT* and *zero-shot* cases) and new relations (*USKT*) which will more likely occur in a low resource setting, which can be best seen by the size of these subsets on the Laptop dataset.

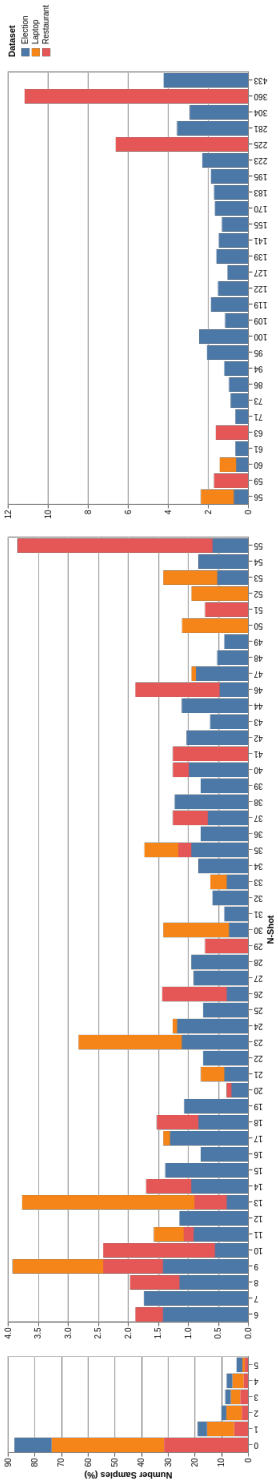


Figure 4.10: Number of samples as a percentage per value of n.

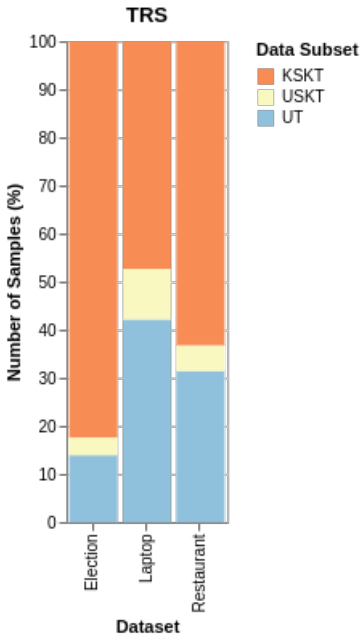


Figure 4.11: Percentage of samples per *TRS* data subset.

	Data Subset		
Dataset	KSKT	USKT	UT
Election	2093	94	354
Laptop	302	67	269
Restaurant	708	60	352

Table 4.11: Number of samples within each *TRS* data subset.

4.2.4 Conclusion so far

Within this subsection the different error splits that exist within the TDSA literature have been reviewed and added too. Furthermore, each of these splits have been analysed on three major datasets showing for the first time how each dataset has different characteristics. The local splits *DS*, *NT*, and *TSSR* show that the Election dataset contains far more unique sentiments per text, whereas the Restaurant dataset contains the most targets per text. The global splits, *n-shot* and *TRS*, find that in the low resource setting a method that can generalise well to new targets and new sentiment relations would be important, which can be seen through the contrast in subset sizes between Election and Laptop datasets. Lastly, table 4.2 summarises the differences in the error splits, examples for each of these splits and subsets can be found in table 4.3. Finally, a summary of all of the statistical breakdowns of each of these splits can be found in table 4.12.

		Dataset		
Data Split	Data Subset	Election	Laptop	Restaurant
DS_i	DS_1	45.8%	83.9%	79.6%
	DS_2	46.5%	14.7%	20.1%
	DS_3	7.7%	1.4%	0.3%
NT	*1-target*	4.0%	40.6%	25.4%
	low-targets	47.9%	32.0%	34.3%
	med-targets	28.7%	15.5%	30.6%
	high-targets	19.5%	11.9%	9.6%
TSSR	*1-TSSR*	4.0%	40.6%	25.4%
	1-multi-TSSR	41.8%	43.3%	54.2%
	high-TSSR	25.8%	5.0%	8.8%
	low-TSSR	28.4%	11.1%	11.6%
TRS	*KTKS*	82.4%	47.3%	63.2%
	USKT	3.7%	10.5%	5.4%
	UT	13.9%	42.2%	31.4%
n-shot	*zero-shot*	13.9%	42.2%	31.4%
	low-shot	29.1%	19.6%	23.8%
	med-shot	29.0%	19.3%	23.0%
	high-shot	28.0%	19.0%	21.7%
Total Samples		2541	638	1120

Table 4.12: Summary statistics of all splits

4.3 Method Performance on the Error Splits

4.3.1 Introduction

In this sub-section the performance of the following four NN based methods will be analysed across the five different error splits stated in section 4.2. The results from these experiments will be used to further analyse what the error splits show as well as create multiple baselines for these splits. The four different methods are the following:

1. *CNN* – The sentence level CNN from Kim (2014) that encodes the context/sentence and does not take into account the target.

2. *TDLSTM* (Tang et al., 2016b) – An LSTM that encodes the target by ensuring the target word(s) are always the last word(s) fed to the LSTM from the context/sentence. Thus a position based method where the position is encoded through the NN architecture. This is the same TDLSTM as that from chapter 3, described in detailed within section 3.3.3.

3. *IAN* (Ma et al., 2017) – Encodes the target into the context via attention.

4. *Att-AE* – A model that is the same as the *AE* model from Wang et al. (2016b) but with an attention layer after the LSTM enocder. This model is also the same as the inter-aspect model (from now on called *Inter-AE*) from Hazarika et al. (2018) but without the LSTM aspect encoder (phase 2 in figure 1) that models other targets from the same context/sentence.

Thus to summarise the differences in the TDSA methods (last three methods from above); *TDLSTM* encodes the position of targets through its architecture, *IAN* encodes the target into the context via attention and also encodes the context into the target through attention, and *Att-AE* encodes the target into the context through concatenation of the target vector onto each word vector within the context before the LSTM encoder. The *Att-AE* does perform attention over the context but unlike *IAN* does not explicitly model the target in the attention of the context nor does it perform attention over the target word(s)[11].

The reason why *Att-AE* is neither exactly *AE* (Wang et al., 2016b) nor *Inter-AE* (Hazarika et al., 2018) is due to not wanting to add inter-target encoding to the baseline models, as within chapter 5 all models are going to have inter-target encoding added. Furthermore as the model will have inter-target encoding added in chapter 5 it will convert *Att-AE* to *Inter-AE* thus making it a standard model from prior literature. The use of different methods from those within chapter 3 is due to the surge in purely NN based TDSA methods of recent years, thus the only methods used within this chapter are NN based.

Due to having a non-target method *CNN*, all of the TDSA methods have a baseline to compare against. Furthermore as all experiments will contain results from at least two TDSA methods to a larger extent the results should generalise to different TDSA NN architectures. In the following sections the thesis will:

1. State the experimental setup of all experiments within this section (section 4.3.2).

2. Explore the differences in performance on the error splits across the methods (section 4.3.3).

[11]For the detailed reader, the *IAN's* attention can be denoted as *general* where as *Att-AE* would be *concat* based on the notation from Luong et al. (2015, §3.1).

4.3.2 Experimental Setup[12]

As stated in the introductory section of this chapter (4.1), the three datasets that are used throughout this chapter are the Election, Laptop and Restaurant datasets. However, unlike the error analysis section (4.2), the standard training split for each of the datasets will be further randomly split into a new training and an additional validation split, of which the size of these splits can be seen in table 4.13. The validation set is required so that the early stopping can be used for all of the NN methods. Furthermore, the validation set would usually be used for more hyper-parameter tuning, e.g. finding the best learning rate etc, but due to compute time this is not the case. Instead we selected the most common hyper-parameters from the literature as detailed in table B.1, it will be stated explicitly within this chapter if these hyperparameters are not used. One default hyperparameter of note is the embedding, of which the 840 billion token 300 dimensional GloVe vector (Pennington et al., 2014) (from now on called GloVe and is called that in table B.1)[13] was chosen, as it is the most common default embedding in the TDSA literature. All text will be tokenised using Spacy and then lower-cased[14]. Lastly, all results reported in this section will be results on the test set and all validation results will be reported in the appendix for reproducibility reasons (Dodge et al., 2019). However if there is a large difference between the validation and test results this will be mentioned explicitly in this section.

	Data Split			
Dataset	Train	Validation	Test	Total
Election	6811 (57.24%)	2547 (21.41%)	2541 (21.35%)	11899
Laptop	1661 (56.29%)	652 (22.09%)	638 (21.62%)	2951
Restaurant	2490 (52.73%)	1112 (23.55%)	1120 (23.72%)	4722

Table 4.13: Number of samples.

Due to the splitting of the training dataset, the error analysis split statistics in section 4.2 will not be identical for the global error splits (*n-shot* and *TRS*) between the train/test and train/validation as they rely on a comparison of train and validation/test. Even though they will not be identical they are relatively similar as shown by table B.2. Furthermore, as the local splits (*DS*, *NT*, and *TSSR*) are only reported for the test set, table B.3 shows them for the validation and test set showing that they are again relatively similar, thus results should be comparable between validation and test sets.

Furthermore, for all of the experiments performed in this chapter each model will have trained/ran on the respective data eight times. Thus allowing for the random seed problem, that is known in NN methods within NLP (Reimers et al., 2017), and to be able to perform statistical significance tests that take into account this problem (Reimers et al., 2018)[15]. Reimers et al. (2018) has shown that by using a minimum of eight runs two

[12]The code to generate table 4.13 can be found in the README at the following `https://github.com/apmoore1/tdsa_comparisons#analysis-of-the-datasets`. The code to generate tables B.2 and B.3 can be found in the following notebook `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/TDSA_Error_Analysis.ipynb`.

[13]This is the same 300 dimension GloVe embedding that is used in chapter 3.

[14]The text was lower-cased as none of the three TDSA methods stated in their original works if they lower-cased the texts or not. The assumption here is that they did.

[15]These significance tests are different to those used in chapter 3, as these for the neural network based

models can be compared with a confidence level of 99% which is equivalent to $p \leq 0.01$ no matter if the scores from those runs comes from a non-normal distribution. The two scoring metrics commonly used in TDSA and will be used in this chapter are accuracy and macro F1, of which only accuracy can be assumed to produce scores originating from a normal distribution and thus can use the more powerful parametric tests (Dror et al., 2018). Therefore following Reimers et al. (2018) for the accuracy scores the significance test used will be the Welch's t-test (parametric test) (Welch, 1947) and for macro F1 the Wilcoxon signed-rank test (non-parametric test) (Wilcoxon, 1945). When comparing two models using these statistical tests for each test the one-tailed version of it will be used as in these experiments the requirement is only to know if one model is better than another. In this and the next chapter 5, a method is defined as the general NN architecture where as the model is defined as the concrete configuration of that method. Thus two models can be different but use the same method, for example the difference would be the word vectors that the two models use. When comparing two models across multiple variables and therefore significance tests a correction procedure is required, as explained in section 3.5.1. In this chapter, the Bonferroni correction procedure will be used where appropriate as in none of our cases can independence be assumed.

4.3.3 Baseline Results[16]

4.3.3.1 Introduction

The baseline results use the four different methods stated in section 4.3.1 as is without any changes to their respective NN architectures. These results will be explored to see whether the intuition behind the error splits as stated in section 4.2.3 is to a degree true.

For clarification, the sentence/text level *CNN* method is trained differently to the TDSA methods due to the fact that it does not model the target within the sentence. Thus instead of training the *CNN* method with potentially the same sentence multiple times with potentially multiple different sentiments as is the case with TDSA datasets[17], the TDSA dataset is converted to a text level dataset. To convert from TDSA to a text dataset each text/sentence can only contain one sentiment, from this two options are plausible; 1. only uses texts that contain one unique sentiment (DS_1 dataset), or 2. use the majority sentiment from the text. These two options were compared of which the detailed results can be found in appendix B.3, of which it was found that the second/later option performed best on 2 of the 3 datasets and all datasets for the accuracy and macro f1 metrics respectively across both validation and test splits. For clarification, when predicting with this text level classification method all targets within the same text will be given the predicted text level sentiment label. This experiment of comparing the two

methods better take into account all runs produced by different random seeds.

[16]All tables and graphs within this section have been generated through the following notebook `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/TDSA_Baseline_Results.ipynb`. The exception to this are the tables generated within appendix B.3, of which there is a pointer in that appendix to the relevant notebook.

[17]This was how the non-target aware models from chapter 3 were trained. Therefore the non-target aware models in some training samples would have been given the same sentence with different sentiments to train on. Even though it may have been better to train them in the ways stated within this section, this was not tested due to compute time. Furthermore it was assumed that they were trained in the same way as their target aware methods as it was not stated in the papers (Vo et al., 2015; Tang et al., 2016b; Wang et al., 2017a) that were reproduced that they were trained any differently.

options of training a text classifier on TDSA data was required as prior works that have shown results for text classifiers never explain how they were trained (Tang et al., 2016a; Wang et al., 2016b; He et al., 2018b; Jiang et al., 2019). For a stronger text classifier baseline one could consider pre-training the text classifier from large sources of annotated data such as Yelp reviews (Tang et al., 2015), Amazon reviews (McAuley et al., 2015; He et al., 2016), or Tweets using distant supervision (Go et al., 2009) for the Restaurant, Laptop, and Election datasets respectively and then fine tune them on the respective TDSA dataset. However this stronger baseline is not considered in this work as we are not looking at transfer learning from other sources of sentiment, but this has been shown to be beneficial for TDSA methods (He et al., 2018b).

4.3.3.2 Overall Results

Figure 4.12 shows the results of the baseline models across both the validation and test splits with the associated tables B.4 and B.5 in appendix B.1. From these results it can be seen that the *CNN* text classification baseline is indeed a strong baseline for the Laptop and Restaurant datasets that the TDSA methods find difficult to beat.

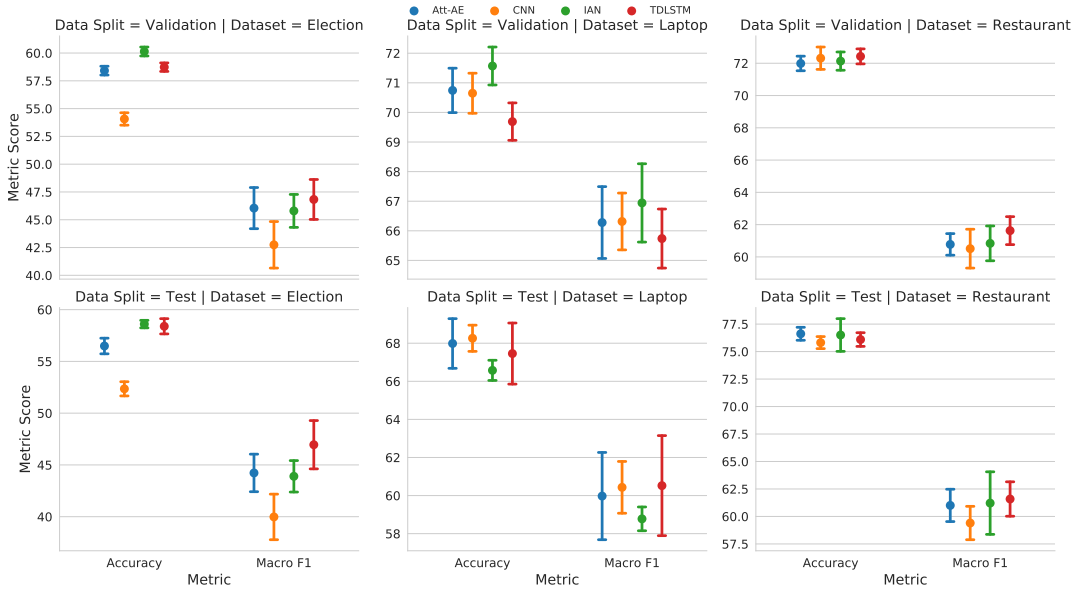


Figure 4.12: The mean and standard deviation error bars from running each model 8 times.

4.3.3.3 Comparison of the Original Model Scores to the Reproduced Models

Figure 4.13 compares the single run performance of the original TDSA models scores from their associated papers to the distribution of eight accuracy scores from our reproduced TDSA methods[18]. As can be seen from figure 4.13 the models original score are within the distribution of scores from the reproduced models apart from *IAN* where the original

[18]Accuracy metric was the only metric reported in all of the original TDSA method papers and none of them reported on the Election dataset.

models performance is a lot higher especially for the Laptop dataset. *IAN's* performance difference is most likely due to the fact that the reproduced version uses a different optimiser, ADAM (Kingma et al., 2015), instead of SGD with momentum (Qian, 1999), this design choice was made so that all models used the same optimiser. Even though it would be good to optimise the performance of the *IAN* model so that it produces scores similar to the original doing so in a fair manner would mean hyperparameter tuning the other models as well (Dodge et al., 2019), which would start becoming computationally expensive. Thus in this thesis it is accepted that the *IAN* model has not been reproduced to the same performance as the original paper reports.

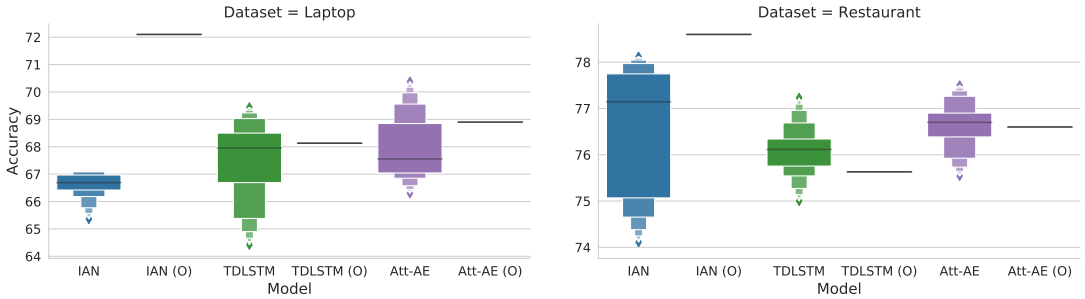


Figure 4.13: Distribution of all scores and the line represents the mean value, of which for the original models this line represents their only reported score. Model names with a *(O)* represent the score reported in the original models paper.

4.3.3.4 Overall Results Comparison between TDSA and Text Classification Models

This thesis shows for the first time that not all TDSA models are statistically significantly better than a text classifier as shown by table 4.14, even though all three original TDSA papers state that their TDSA models are superior to a text classifier. Furthermore, at the 95% confidence level none of the TDSA methods are significant on the Laptop test split no matter what the metric is. This shows that potentially hyperparameter tuning is very important to get the most out of the TDSA models. More likely the reason for the text classifier's strong performance on the Laptop and Restaurant dataset compared to the Election dataset is that these datasets contain a large quantity of DS_1 samples (see figure 4.1), of which it is shown later in this section in figure 4.14 that the text classifier does at least as good if not better than the TDSA models on the DS_1 subset in the Laptop and Restaurant datasets. This further shows that the overall metrics tell us very little in what the difference is between a text classifier model and the TDSA models.

			Dataset		
Model	Split	Metric	Election	Laptop	Restaurant
Att-AE	Test	Accuracy	**2.14e-08**	0.68	**0.01**
		Macro F1	**7.86e-04**	0.67	**0.03**
	Validation	Accuracy	**2.57e-10**	0.40	0.84
		Macro F1	**3.76e-03**	0.52	0.31
IAN	Test	Accuracy	**2.21e-10**	0.99	0.14
		Macro F1	**1.02e-03**	0.99	0.08
	Validation	Accuracy	**3.98e-12**	**0.01**	0.69
		Macro F1	**3.96e-03**	0.16	0.30
TDLSTM	Test	Accuracy	**1.29e-10**	0.87	0.19
		Macro F1	**2.53e-05**	0.46	**0.01**
	Validation	Accuracy	**1.26e-10**	0.99	0.36
		Macro F1	**8.07e-04**	0.85	**0.03**

Table 4.14: The P-values for each model where the null hypothesis is that each model performs as well as a *CNN* text classifier. The P-Values in bold are those ≤ 0.05.

4.3.3.5 Error Split Results

The performance of all of the models across all datasets for each split and their associated subsets can be seen in figure 4.14 (appendix B.2 figure B.1 shows the validation split results). The figures that contain subset performance results will not contain results for the DS_3 subset for the Laptop and Restaurant datasets, this is due to the subset containing very few samples as highlighted in subsection 4.2.3. The error split results for the test and validation splits are better highlighted in figures 4.15 and B.2 respectively where the accuracy on the whole dataset (overall accuracy) is subtracted from the error subset accuracies. These figures can thus evaluate the error splits that were discussed and created within this section. In the list below the results will be analysed by error split:

- *DS* split, as expected, increases in difficulty as the number of unique sentiments in the text increases, thus showing that target sentiment relation to be a difficult task for the models to perform. Furthermore, it can be seen that on average the *TDLSTM* model performs consistently well on the DS_2 and DS_3 subsets compared to the other models.

- *NT* split does not have a consistent affect on the performance of the models, this was also found in one of the original papers (Zhang et al., 2019a). One would expect texts that contain a lot of targets to be more difficult and at times it is as shown by the Restaurant dataset. However, on the other two datasets this is not the case. Furthermore, the performance across the subsets can differ between datasets splits, e.g. the performance of all models on the *med-targets* subset on the Election dataset is worse than *high-targets* for all models on the test split, but on the validation figure (B.2) the opposite is true. This would suggest that even though theoretically a text with more targets should be more difficult for a model to classify, due to the complexities of matching targets to their respective

sentiments (Zhang et al., 2019a), this is not the case. Thus later in this section, analysis is conducted to investigate what other factors are influencing the change in performance within the *NT* split. However, for now it can be concluded that *NT* cannot effectively evaluate the target interaction as no consistent trend can be found in this split.

- *TSSR* As expected the *1-Multi* subset is by far the easiest subset to classify suggesting that the models are exploiting the fact that all targets have the same sentiment. The *1* subset tends to perform the next best with the *high* subset at times quite close if not the same. A reason for the *high* subset to have such high performance across the models could be due to the models overfitting to the most frequent sentiment class in the text, as suggested in section 4.2. As expected the *low* subset is by far the worst across all datasets and models and in some cases harder to classify than samples within DS_2 and DS_3. Only on the Laptop test split are the *high* scores similar to the *low*, of which this might be due to the lack of samples for the *high* subset (5% of the dataset) compared to the (11.1% of the dataset) in the *low* subset, which is also suggested by the large error bars. Furthermore the sentiment overfitting which this split is supposed to measure does show to some extent where the TDSA model, *TDLSTM*, that performs consistently better or at least as good in the DS_2 and DS_3 subsets tends to have a smaller difference between subset *1-Multi* and *1*, and is consistently a lot higher than the text classifier on the *low* subset. However this split does not measure sentiment overfitting explicitly very well without the text classification baseline and the *DS* split. For example without *DS* and the text classification baseline it would be impossible to know that the *TDLSTM* is performing target sentiment relation well on the laptop dataset as the DS_2 subset performance could be high due to *TDLSTM* predicting the most frequent sentiment class. This cannot be the case as the performance of *TDLSTM* on the *high* and *low* *TSSR* subsets are both above the text classification model unlike the other two TDSA models. Though this is a rather loose way of measuring sentiment overfitting and is not the way that was stated in the previous subsection 4.2.3. In the previous subsection 4.2.3 the difference between the *high* and *low* subsets was hypothesised to indicate sentiment overfitting, but as can be seen from the figures *TDLSTM* that is supposed to not be overfitting as much as the other TDSA models does indeed contain a low difference between *high* and *low* on the Laptop dataset, but so does *IAN* thus making the hypothesis less likely to be true. Therefore to conclude on the *TSSR* split, it cannot measure sentiment overfitting nor would it be able to measure target interaction as suggested in 4.2.3 either as it would be impossible to know if it was target interaction or sentiment overfitting. However there are clear signs that the subsets measure to some degree target sentiment relation as the score of subsets *1*, *high*, and *low* are similar in order to subsets DS_1, DS_2, and DS_3 respectively and these subsets co-occur frequently as shown in figure 4.7. Thus after this subsection the *TSSR* split will no longer be used.

- *TSR* again the finding is expected where the *USKT* is by far the most difficult subset. The *UT* is more difficult in general than the *KSKT* but with a much smaller margin. This finding is therefore in line with the relation extraction literature

where unknown entities are easier to predict than unknown relations (Levy et al., 2017; Abdou et al., 2019). Within the validation results for the Election dataset the margin between *UT* and *KTKS* is very small. This very much suggests that the models do require a certain amount of supervision for all targets in all sentiment classes or else they bias the target more towards one sentiment class than another. This type of bias can be very harmful as shown by the *USKT*. This could suggest a reason why the margin between *UT* and *KSKT* is so small as some of the *KSKT* targets might not occur in enough samples within a sentiment class. Furthermore the *KSKT* subset can be seen as the upper limit for the other two subsets as it can be seen as the data rich subset.

- *n-shot* the expected result can be clearly seen in all the datasets within the test split but less so within the validation split. Where the expectation is that the greater n is the easier the subset will be. Within the validation split the Election and to some extent Restaurant datasets are the major outliers, where no matter what the subset is, the scores are almost all the same. A reason for this could be that the validation split is used in early stopping therefore some information is leaked to the model. As both the *n-shot* and *TSR* splits measure a model's generalisation to new targets, from the results shown it would appear that *TSR* does this more explicitly. The *TSR* split unlike the *n-shot* models both the unseen targets and unseen relations, of which modelling both has been shown through *TSR* to be crucial. This finding creates another possible explanation why the *n-shot* subsets do not always show a positive correlation between n and the metric score. Furthermore, the *TSR KSKT* subset is always the best performing subset within the split unlike the *high* in the *n-shot*. Thus, for exploring a model's ability to generalise to unknown targets and unknown sentiment relations *TSR* is recommended compared to *n-shot*. Thus, like the *TSSR* split the *n-shot* will not be used after this subsection.

Generally, the test and validation results from figures 4.14 and B.1 respectively show that the *DS*, *TSSR*, and *TSR* splits contain the most difficult subsets. The TDSA models perform a lot better on the DS_2 subset on the Election datasets compared to the Laptop and Restaurant datasets. This could be due to the Election dataset containing far more DS_2 samples relative to it's overall size compared to Laptop and Restaurant datasets (see figure 4.1). This may suggest that ways to improve the models performance on the DS_2 and potentially DS_3 subsets could be by training the models on more of these samples and thus improving the target sentiment relation modelling. However, this could have a negative affect on the performance in the DS_1 subset. Also how to generate more DS_2 and DS_3 samples could also be a difficult and interesting challenge.

From the test and validation results in figures 4.14 and B.1 respectively the text classification model, has a few unexpected findings. The *TSR*, and *n-shot* splits do not explicitly probe a models capability to model the target sentiment relationship rather how well a model generalises to new targets or less seen targets and unknown sentiment classes for known targets. These probes thus do not explicitly require target information, for example in the *DS* split for the DS_2 subset without modelling the target it is impossible to get all the samples correct, this is not directly true for the subsets in the *n-shot* and *TSR*. However, as can be seen from the results the text classification model does not perform equally well across all subsets in the *n-shot* and *TSR* splits, this suggest

that either the text classification model does use the target to influence the sentiment prediction, or these subsets correlate with other dataset factors, for example *zero-shot* subset has far fewer samples that belong to the DS_2 subset than the *high-shot* subset. These issues are not explored any further but should be looked at in the future.

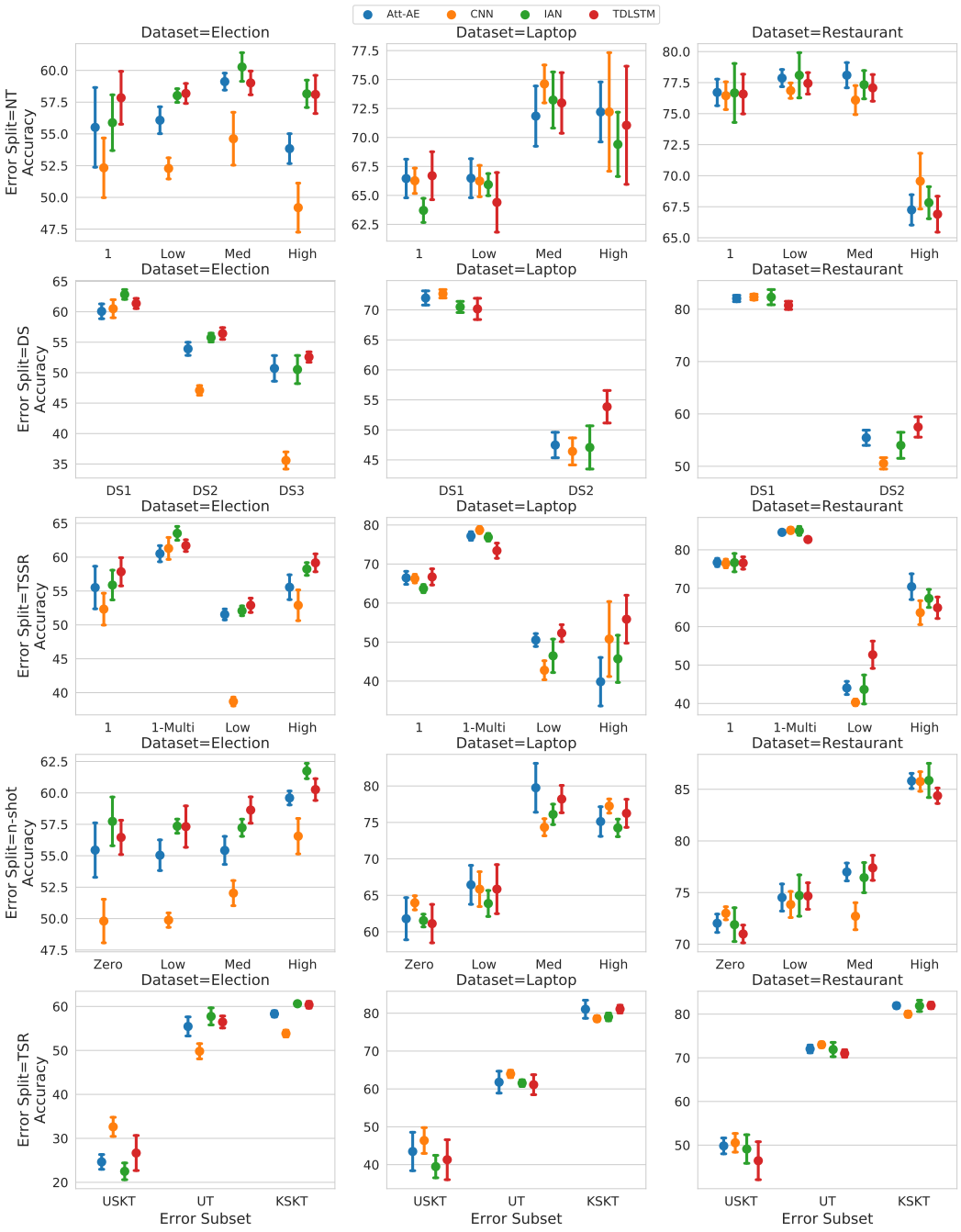


Figure 4.14: The mean and standard deviation error bars for each error subset within all of the error splits on the test split across all datasets.

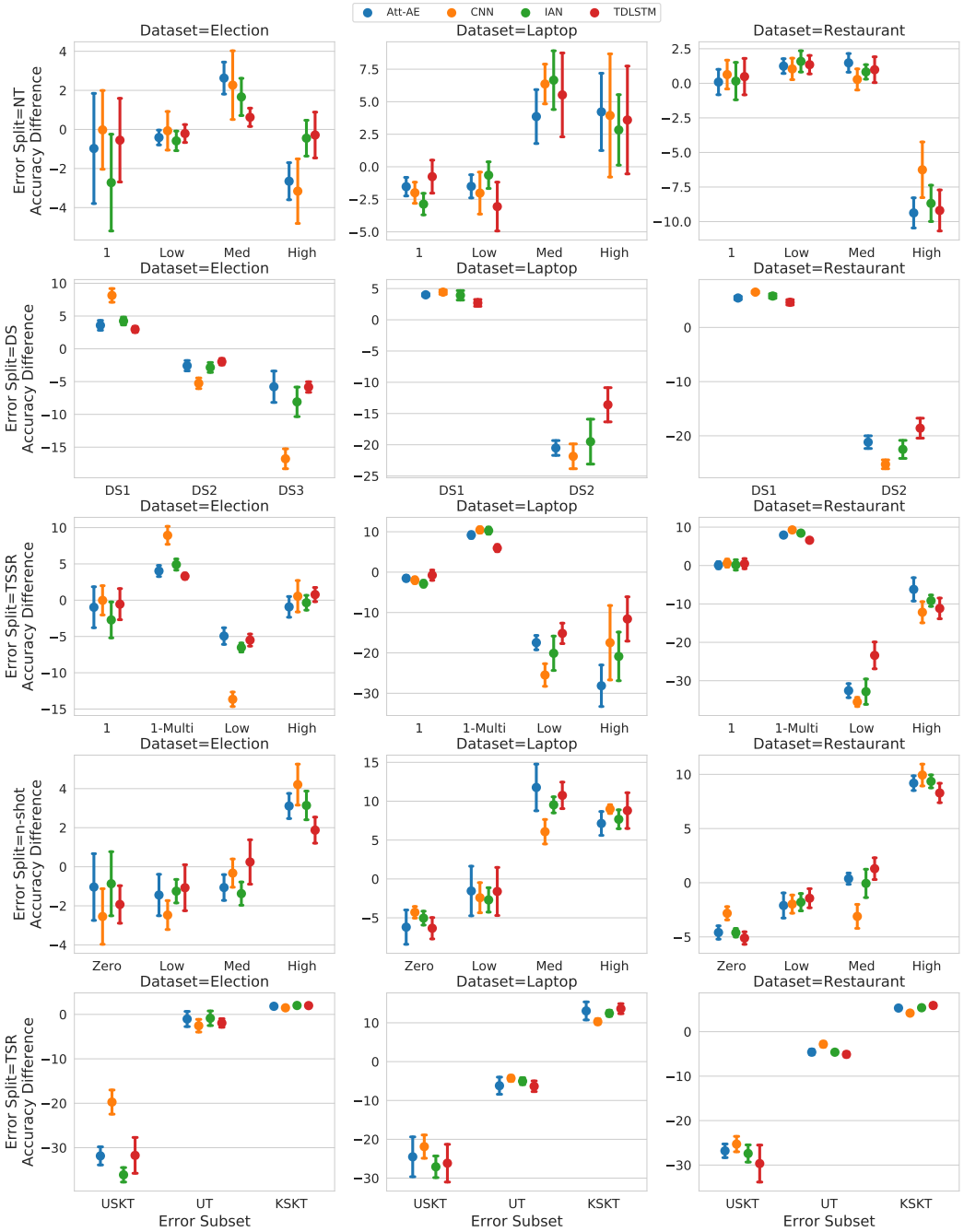


Figure 4.15: The mean and standard deviation error bars for the difference between the overall accuracy and the accuracy from each error subset within all of the error splits on the test split across all datasets.

4.3.3.6 Error Split Results Comparison between TDSA and Text Classification Models

Using the test and validation results from the subsets shown in figures 4.14 and B.1 respectively it is possible to explore the differences between the text classification model and the TDSA models. These differences can be better seen through the heatmaps in figures 4.16 and 4.17, where the former is not corrected for multiple significance tests where as the later is using Bonferroni and is aggregated across datasets. Note that for figure 4.16 the DS_3 subset results should be ignored for the Laptop and Restaurant datasets as they were never calculated as the sample size for the DS_3 subset is too small. Also the DS_3 subset is removed from figure 4.17 as only the Election dataset contains enough samples to create confidence scores. From all of these figures it is clear to see that the subsets that the TDSA models outperform the text classification model in are DS_2, DS_3, *low-TSSR*, *TSR KSKT*, *n-shot Med*, and *NT Low*. There are other subsets where the difference is significant as shown in the heatmaps but the majority of these significant differences only occur because of the Election dataset as can be seen if you compare figures 4.16 and 4.17. Furthermore, the outliers in these differences are the *n-shot Med*, and *NT Low* subsets of which the reason why it is believed these are outliers was described in the previous paragraphs. The DS_2, DS_3, and the *low-TSSR* are expected to perform better for the TDSA models as they contain multiple unique sentiments within a sentence, for which a text classification model can only predict one of those sentiments for the sentence thus, limiting the model's capability to perform well on these subsets. This therefore shows that the TDSA models must be learning some target sentiment relationship modelling or else they would not be more competitive than the text classifier. The *TSR KSKT* shows that when the TDSA models have seen a target enough times in a known sentiment context then they can perform a lot better than the text classification model and their respective overall accuracy. However it is the other subsets within *TSR* that are of more interest showing the deficiencies of the TDSA models. The worse subset within *TSR* is the *USKT* of which this is the only subset where the text classification model in general perform significantly better (see figure 4.17). TDSA models are most likely biasing the target representation towards a subset of sentiment classes for those targets and hence why the text classification models perform better on those targets. The *UT* (*n-shot zero* is the same subset) subset is an interesting result as it is dataset dependent as shown best in 4.16, in the Election dataset the TDSA models are better but in all other datasets the text classification model is better. This could be due to the size of the datasets as the Election dataset is much larger than the rest and therefore could allow the TDSA models to create better general target representations, thus allowing the models to leverage similarities with known targets. From the dataset heatmap figure 4.16 it can be easily seen that on the Election dataset the majority of subsets are statistically significant compared to the Laptop and Restaurant dataset. This is most likely due to the Election dataset containing more targets per text (as can be seen in table 4.15 [19]) and therefore far fewer texts within DS_1 which is the subset the text classification model is most suited to. Even though the text classifier does not perform statistically significantly better than all the TDSA models on the Laptop and Restaurant datasets for DS_1 they are never worse. Furthermore, as the

[19] Dataset statistics for the splits, rather than the whole dataset, used in this section can be seen in table B.6.

Laptop and Restaurant datasets are mainly made up of DS_1 samples (see figure 4.1) this is most likely the reason why the TDSA models are not statistically significantly better than the text classification models on these datasets.

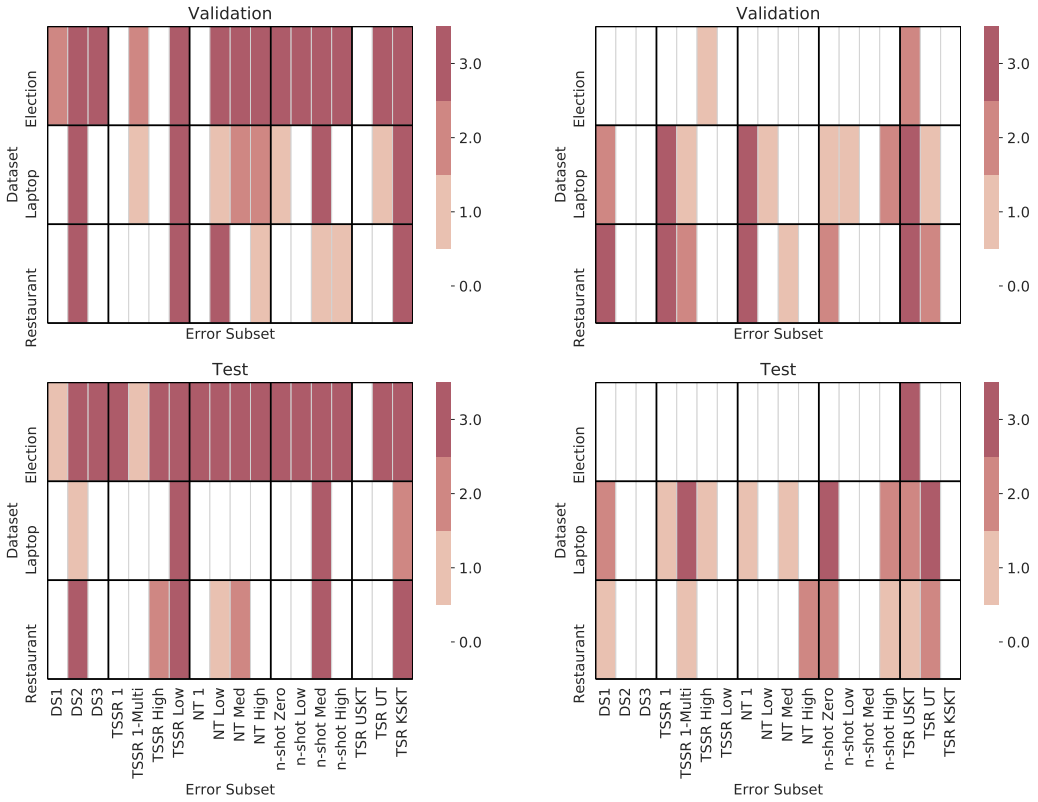


Figure 4.16: Plots in the first column represent the number of TDSA models that are statistically significantly better than the text classification model. Plots in the second column show the opposite, the number of TDSA models where the text classification is statistically significantly better. All plots have a confidence level of 95% ($p \leq 0.05$).

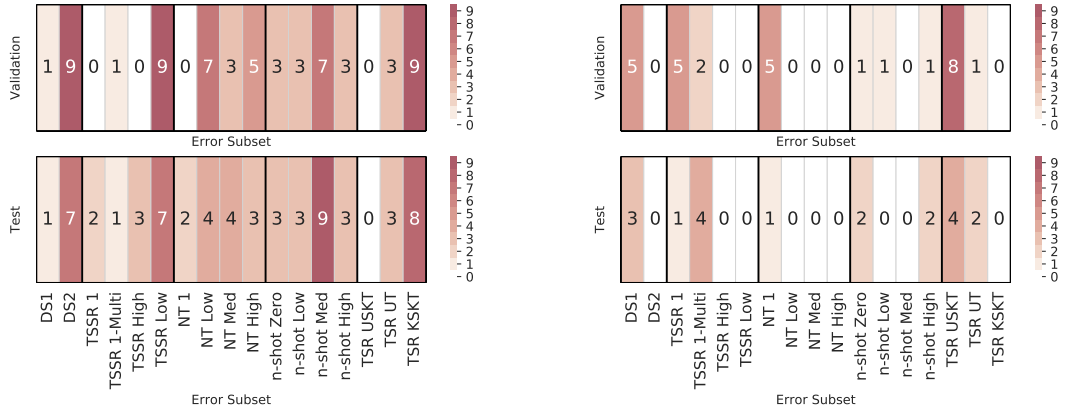


Figure 4.17: Plots in the first column represent the number of TDSA models that are statistically significantly better than the text classification model across all datasets. Plots in the second column show the opposite, the number of TDSA models where the text classification is statistically significantly better. All plots have a confidence level of 95% ($p \leq 0.05$) and have been corrected using Bonferroni.

Name	No. Sents(t)	No. Targs (Uniq)	ATS(t)	POS (%)	NEU (%)	NEG (%)
Laptop	1872	2950 (1181)	1.58	1328 (45.02)	628 (21.29)	994 (33.69)
Restaurant	2578	4722 (1528)	1.83	2892 (61.25)	829 (17.56)	1001 (21.2)
Election	4045	11899 (2179)	2.94	1744 (14.66)	4572 (38.42)	5583 (46.92)
No. Sents(t)=number of sentences that contain a target, No. Targs (Uniq)=Number of (unique) targets (all targets are lower cased), ATS(t)=Average Target per Sentence where the sentences must contain a target, LABEL (%)=Number of LABEL samples (percentage of LABEL samples).						

Table 4.15: Dataset statistics for each datasets where each dataset represent the combination of all the dataset's splits e.g. train, validation, and test.

4.3.3.7 Comparing the Error Split Results to the Prior Work

From the results in this subsection the findings can relate back to some of the original work on these splits confirming the same findings. The findings of the *TSSR 1-Multi* performing better than *TSSR 1* is the same finding as Nguyen et al. (2015) as both of these subsets are the equivalent to *ST2* and *ST1*. When the number of unique sentiments increase in a text, which can be measured through the *DS* and *TSSR* splits, this reduces the performance of a method, which is the same finding as Xue et al. (2018) comparing the normal test to the *hard* test, and that of Nguyen et al. (2015) comparing *ST1* or *ST2* with *ST3*.

The *n-shot* findings do not confirm the findings of the original work by Yang et al. (2018) where they found that in general models performance do not correlate with the number of times the target/aspect appeared in the training data. However, our findings show that they do correlate where the more the target appears in the training data the better the performance in general. The reason for the difference could come from the task itself, as Yang et al. (2018) was not solving the task of TDSA but rather the task of Multi-Entity Aspect Based Sentiment Analysis (ME-ABSA), where TDSA would be equivalent if when predicting the sentiment of the target the latent aspect was also given. This difference in task could make a large difference as knowing a target's latent aspect could greatly improve a model's performance on unknown targets. The reason why the latent aspect would make such a large difference is because the dataset would contain a few aspects which occur frequently therefore allowing the model to create a good representation for the aspects. Furthermore, given these aspects the likelihood is that there could be many unknown aspect target pairs but due to the model potentially having a good representation of the aspect the performance on these *zero-shot* pairs could be quite high. Thus, a reason for Yang et al. (2018) finding no correlation between performance and number of times the target/aspect appeared in the training data could be due to the aspects.

As found earlier in figures 4.14 and B.1 for the test and validation results respectively the *NT* split does not show any consistent trend, which to some degree is what is found in the original works (He et al., 2018a; Zhang et al., 2019a). The expected trend was as the number of targets increase the lower the performance. A potential reason for this could be that there are other factors that influence the *NT* split. The factors that will be explored here are the target sentiment relationship factors which can be measured to some extent using the *DS* and *TSSR* splits. To explore this all the datasets will be first subsetted by one of the *DS* or *TSSR* subsets and then further subsetted by one of the *NT* subsets, the model's performance will be measured on each one of these compounded subsets. Figures 4.18 and 4.19 show the performance on these compounded subsets whereby the former subsets the data by *DS* and the latter *TSSR*, for the validation data this can be seen in figures B.3 and B.4. Note that in the figures some of the *NT* subsets do not exist on the x-axis, this is because after the subset compounding no data exists for those subsets. These figures show that in general for the DS_1 and *TSSR 1-Multi* rows the larger *NT* the better the performance of the models, of which for the DS_1 row this can be better seen in the validation data (figure B.3) than the test. This is most likely the case because of the models exploiting the fact that there are more targets expressing the same sentiment. This exploitation of targets expressing the same sentiment can also be seen in the DS_2 rows (better seen in the validation data) where the more targets the

better the score. Within the *Low* and *High TSSR* subsets the trend is less clear. The expectation within these subsets would be, when there are more targets (larger NT) this will results in poorer performance for the *Low TSSR* subset, but better results for the *High TSSR* subset. This expectation is under the assumption that the more targets there are within the *High TSSR* subset the greater the likelihood that the targets have the same sentiment and exploiting the most frequent sentiment would gain a higher performance score. The opposite is true when there are more targets within the *Low TSSR* subset the greater the likelihood that the targets have a different sentiment and exploiting the most frequent sentiment would gain a lower performance. However, this expectation is not always true and can be inconsistent between splits, for example the *High TSSR* Laptop results have different trends between test and validation splits. Furthermore, this assumption of more targets within the *TSSR Low* and *High* subsets does not necessarily mean more targets of the most frequent sentiment class due to the way *TSSR* subsets are created (equation 4.1), hence a potential reason why there is no consistent correlations in those subsets. Thus, this analysis shows to some extent why the NT split has no trend as the target sentiment relationship factors are more influential than the number of targets on the performance of the models. This therefore solves to some extent why Zhang et al. (2019a)[20] and He et al. (2018a)[21] also could not find a steady trend for the NT split.

The DS_i split was stated to get more difficult as i increased and this has been shown in this work and in the original (Wang et al., 2017a). However, as mentioned in section 4.2.1, in the original work it was also shown for some methods and metrics that the models perform best on the DS_3 subset. In this work that phenomena did not occur when using the accuracy metric, which Wang et al. (2017a) did not use. Therefore, to test if the DS split results do change because of the metric, in figure 4.20 are the DS results on all datasets using the macro F1 metric which was one of the metrics used by Wang et al. (2017a). As can be seen from the results the only dataset where i in DS_i does not negatively correlate with the macro F1 results is the Election test dataset. The Election test datasets was also the only dataset Wang et al. (2017a) used when measuring model performance on the DS split[22]. The other results follow the trend shown in this section when using the accuracy metric.

To investigate why the Election test dataset does have a different trend when using a different metric the approach taken was to explore the individual F1 scores for each sentiment label on the Election dataset. This approach was taken as the main difference between accuracy and macro F1, as the macro F1 is not biased by the un-balanced label distribution that is within the dataset, of which all datasets used are very un-balanced (see table 4.15). Figure 4.21 shows the results of each sentiment label's F1 score and as expected the most frequent sentiment (negative) has the highest scores no matter the subset. These results highlight why the macro F1 score does not have the same negative correlation as the accuracy metric does for the DS split. As the results show the positive and the neutral sentiments do not follow the negative correlation that the negative sentiment does. Thus, as the scores of each sentiment are weighted the same in the macro F1 metric this therefore causes the macro F1 score for the DS_3 subset to be higher than the DS_2 subset in the test split. The potential reason for this unusual correlation could

[20] See figure 4 in the paper.
[21] See figure 3 in the paper.
[22] Results can be found in table 4 of Wang et al. (2017a).

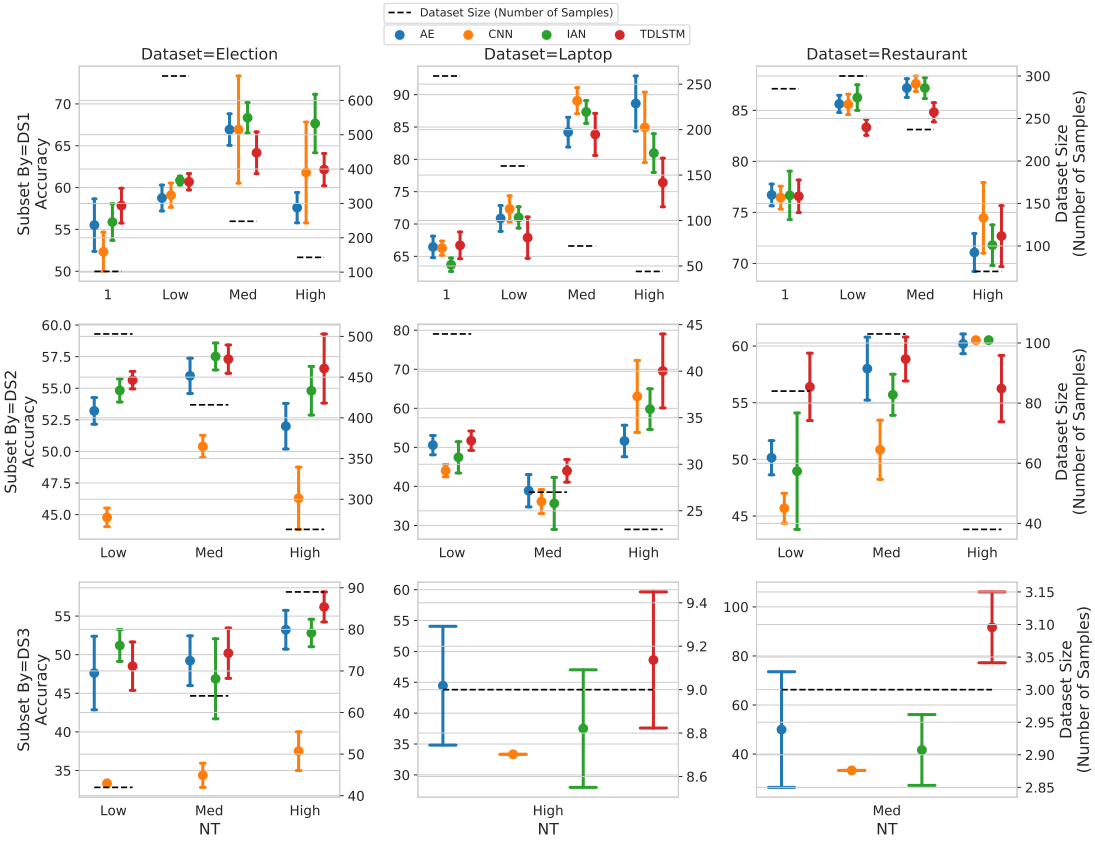


Figure 4.18: Each plot shows the performance (y-axis accuracy) of the given models and sample size of the data evaluated on (y-axis dataset size) on the different test datasets (columns) after being subsetted by the relevant *DS* subset (rows) and then NT subset (x-axis).

be due to the model overfitting to the most frequent sentiment class (negative) and hence why if the model predicted negative for all samples in a DS_3 sentence then it would get some samples correct but it would get at least 2 samples wrong.

To further investigate whether the most frequent sentiment class always has this negative correlation the results for the Restaurant and Laptop datasets are shown in figures 4.22 and 4.23 respectively. From these two plots we can see that the most frequent sentiment class (positive for both datasets) has the largest drop in F1 score from DS_1 to DS_2. This large drop in F1 score gives some extra merit to the idea that the models are overfitting to the most frequent sentiment class[23]. Due to this overfitting the model is most likely predicting the most frequent sentiment class more often than it should where as in the cases for the least frequent sentiment classes it could be only predicting these when it is confident. These reasons are not empirically proven but the results have shown further insight into the *DS* split. Lastly, these results show more that the results from the original paper (Wang et al., 2017a) do not generalise across datasets and that the general result is that the metrics normally correlate negatively with the *DS* subsets.

[23]It was also found in chapter 3 section 3.6 that NN based methods overfit to the most frequent

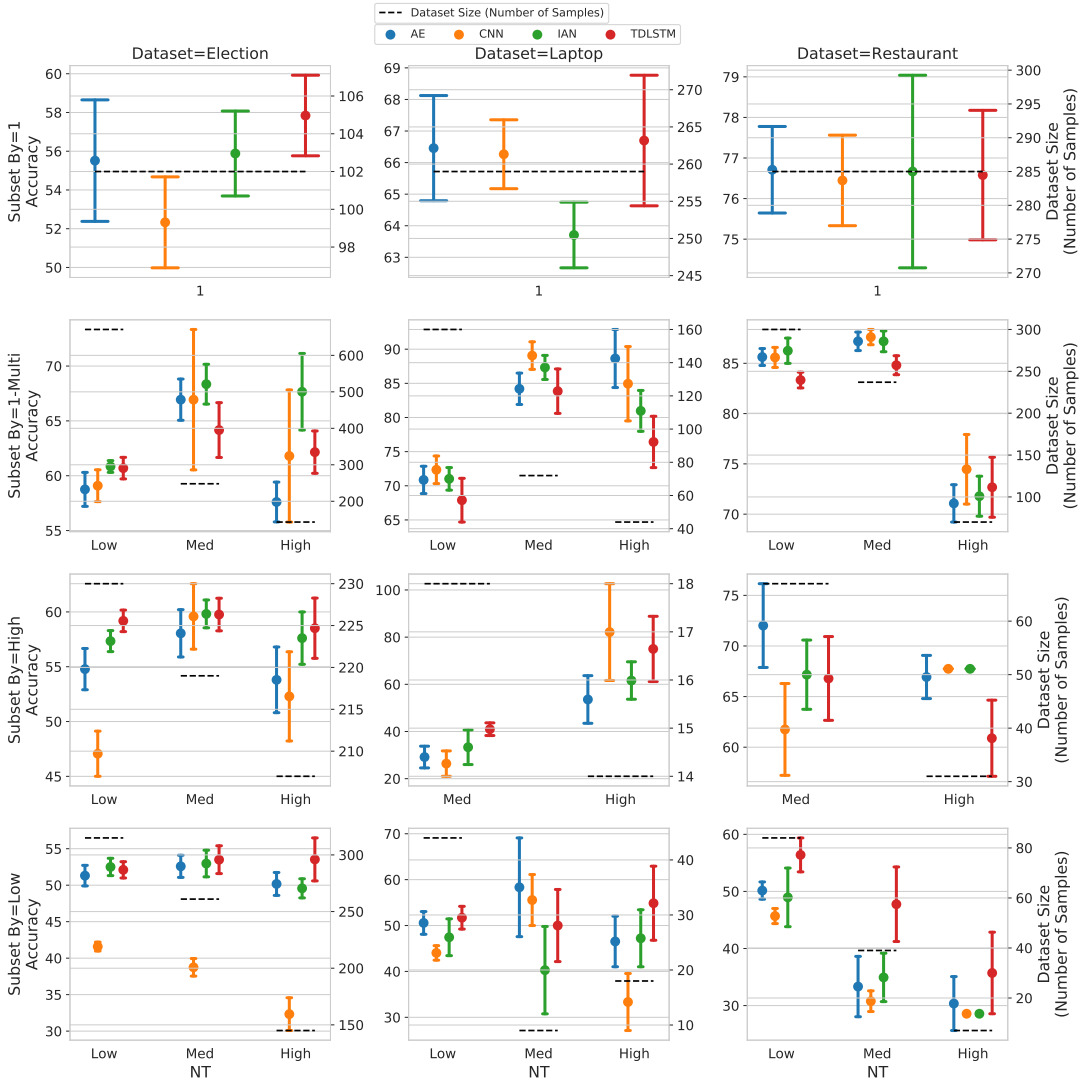


Figure 4.19: Each plot shows the performance (y-axis accuracy) of the given models and sample size of the data evaluated on (y-axis dataset size) on the different test datasets (columns) after being subsetted by the relevant *TSSR* subset (rows) and then NT subset (x-axis).

sentiment class on many datasets in a low resource setting.

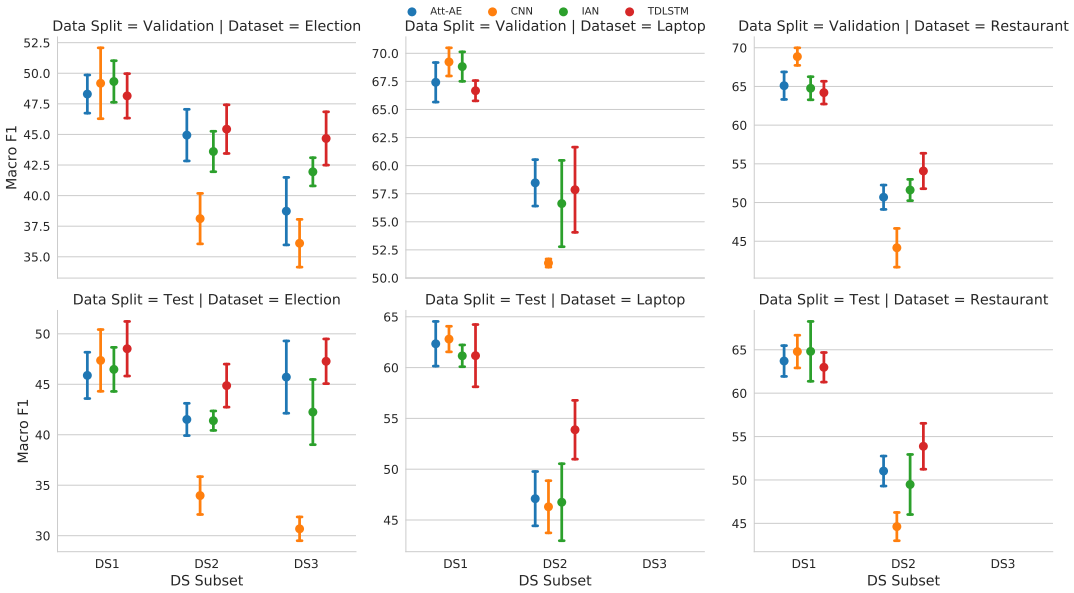


Figure 4.20: Macro F1 score where the data is subsetted by the *DS* split.

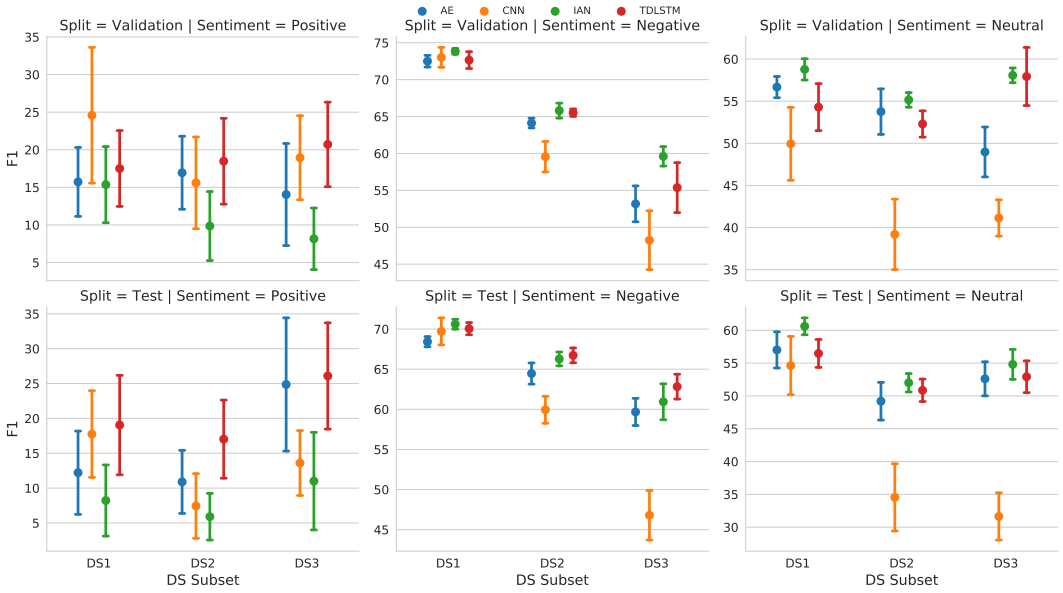


Figure 4.21: The Election test and validation split F1 scores for each sentiment label, where the data has been further broken down through *DS* subsets.

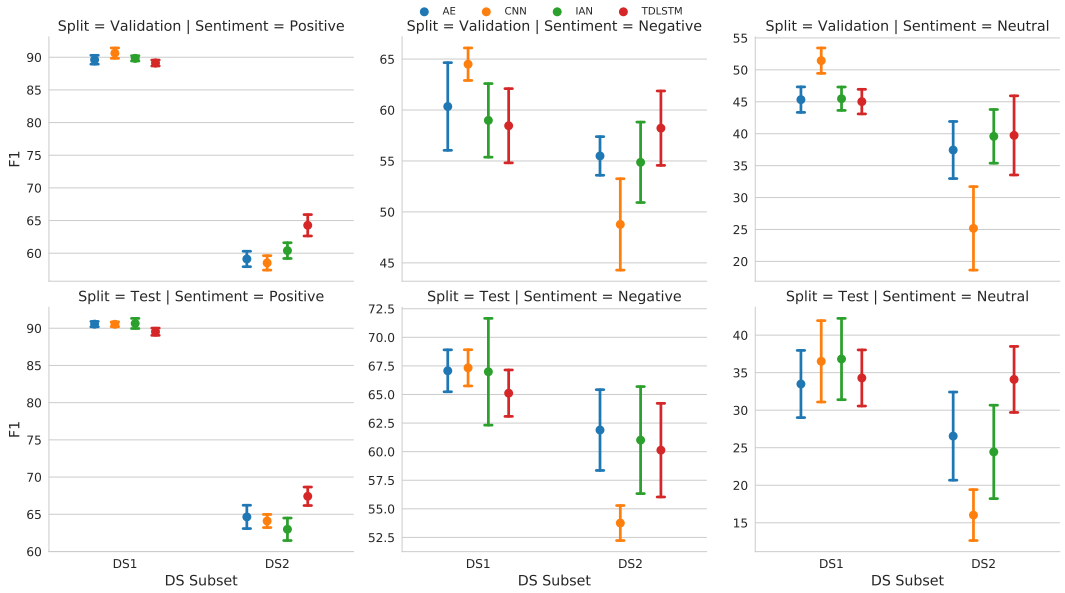


Figure 4.22: The Restaurant test and validation split F1 scores for each sentiment label, where the data has been further broken down through *DS* subsets.

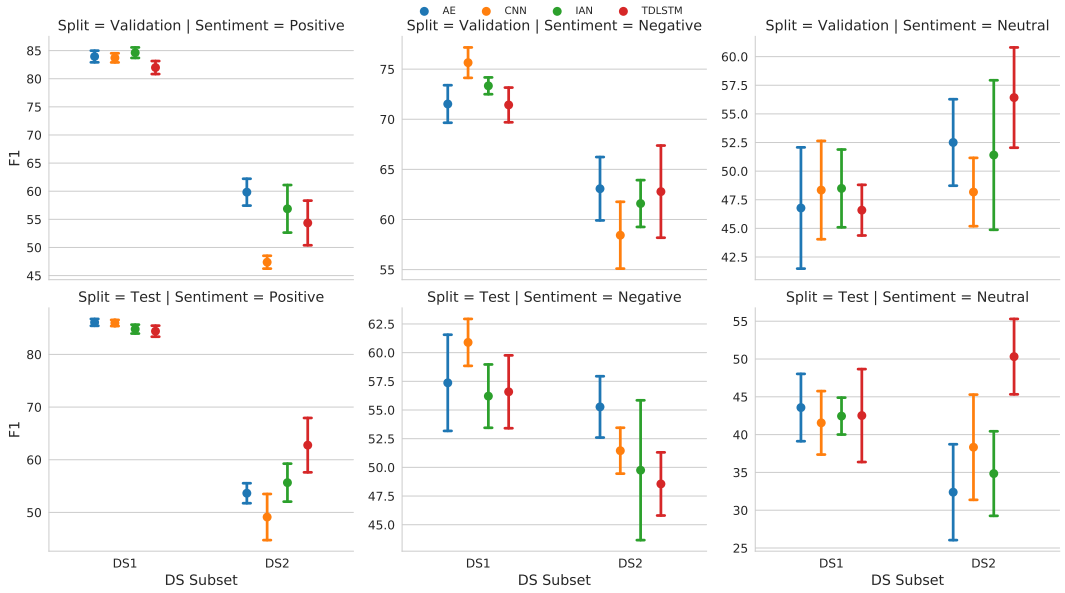


Figure 4.23: The Laptop test and validation split F1 scores for each sentiment label, where the data has been further broken down through *DS* subsets.

4.3.4 The Strict Text ACcuracy (STAC) Metric

Both the *DS* and *TSSR* error splits explore the concepts of target sentiment relationships and overfitting to the most common sentiment within a text. However, neither of these can create error subsets that will explicitly inform you if the model can detect sentiment for all the targets in the text and thus performing the target sentiment relationship task perfectly. Both the *DS* and *TSSR* splits do attempt to show this but both are subject to the model finding the most frequent or easiest to find sentiment for some/all of the targets in the subsets. Thus, the creation of the Strict Text ACcuracy (*STAC*) metric. This metric works on the sentence/text level compared to the accuracy and macro F1 metrics that are based at the target level. *STAC* treats each sentence as a sample and each sentence can only be correct if all targets within that sentence have been classified correctly, it is then averaged by the number of sentences. The *TAC* equation 4.2 that is used within the *STAC* equation 4.3 finds the average number of targets that are correct within a sentence. The notation to describe *STAC* and *TAC* in equations 4.3 and 4.2 is the same notation used in equation 4.1, which describes the *TSSR* split. T_j within *STAC* represents all of the targets within sentence j from all sentences X that is within the dataset, and t_{ji} represents target i true sentiment within sentence T_j where $\hat{t_{ji}}$ is the predicted sentiment.

$$\text{Text ACcuracy (TAC)}(T_j) = \frac{\sum_{i=1}^{|T_j|}[t_{ji} = \hat{t_{ji}}]}{|T_j|} \tag{4.2}$$

$$\text{Strict Text ACcuracy (STAC)} = \frac{\sum_{j=1}^{|X|} \begin{cases} 1, & \text{if TAC}(T_j) = 1 \\ 0, & \text{otherwise} \end{cases}}{|X|} \tag{4.3}$$

The *STAC* metric is more useful when applied to subsets of a dataset, thus two specific versions of the *STAC* metric are created:

1. *STAC 1* - The *STAC* metric applied to only the data in the DS_1 subset.
2. *STAC Multi* - The *STAC* metric applied to only the data in the DS_2 and DS_3 subsets.

The *STAC Multi* gives in one metric how well overall a TDSA model is at target sentiment relation modelling removing all factors of overfitting to a sentiment class, or predicting the most frequent sentiment, of which this is possible in *STAC 1*. *STAC Multi* can also be seen as a coarse grained and much stricter version of the *DS* split, as both measure target sentiment relationship modelling. However, due to *STAC Multi* being such a strict and thus difficult metric the *DS* subsets can be useful to measure target sentiment relationship modelling at a more fine grained scale. For example, if a model does not perform significantly better nor worse than another on *STAC Multi* but does perform better on DS_2 and DS_3 subsets, then the likelihood is that the model is performing target sentiment relationship modelling better. The difference between *STAC 1* and *STAC Multi* can show to some degree how much the model is overfitting to the most frequent sentiment class in a text. The performance of all models across all metrics including accuracy and macro F1 can be seen in 4.24, the *STAC* metric is shown for completeness. As can be seen from the figure the *STAC Multi* is by far the most difficult

metric and scores much lower than any of the accuracy metrics on any of the subsets shown in figure 4.14 (validation split figure B.1). However, the *STAC 1* results can be the easiest metric as shown by the Restaurant dataset. The difference between *STAC 1* and *STAC Multi* for the TDSA models is rather large and more so for the Restaurant and Laptop datasets which could be due to the fact there are proportionally and overall more DS_2 and DS_3 sentences in Election than the Restaurant and Laptop datasets as shown by table 4.16. Furthermore, as should be the case, the text classification model (*CNN*) scores 0 in all of the *STAC Multi* thus showing again the point of the metric and the relevancy to TDSA. These scores highlight that TDSA models have much to improve upon with regards to target sentiment relation modelling as shown by the *STAC Multi* metric without resorting to simpler majority sentiment classification of the sentence as shown by *STAC 1*, and the other error subsets (DS_1, *TSSR 1*, and *TSSR 1-Multi*). Furthermore, from the results it is interesting to see that the *TDLSTM* model generally performs well on the *STAC Multi* metric compared to the other models across all datasets, of which this could be due to the model encoding position of the target within its architecture.

Figure 4.24: Performance across all metrics for all models across all datasets and splits.

		Subset		
Split	Dataset	DS_1	DS_2 and DS_3	Total
Train	Election	1227	1092	2319
	Laptop	933	118	1051
	Restaurant	1162	216	1378
Validation	Election	467	396	863
	Laptop	364	47	411
	Restaurant	497	103	600
Test	Election	469	394	863
	Laptop	373	38	411
	Restaurant	520	80	600

Table 4.16: Number of sentences in each split for all datasets.

4.4 Reflection on Error Splits, STAC, and the Results

Through these error splits and new metrics (*STAC*) the differences between the TDSA and text classifier can be seen and where the TDSA models do outperform the text classifiers by a large margin. The flip side to distinguishing the performance of TDSA and text classifier models is by creating 'challenge datasets' that examine the performance of a model in specific circumstances. This approach was taken by Jiang et al. (2019) where they created a new version of the Restaurant dataset called Multi-Aspect Multi-Sentiment (MAMS). The MAMS dataset as the name suggests only contain texts that have at least two targets with at least two different sentiments, thus removing all texts that only have one sentiment. This new dataset was created to avoid samples being easily classified by a text level classifier. This dataset therefore fits into the DS_2 and DS_3 only subsets from the *DS* split. They found a large difference in scores between the text classifier models and the TDSA ($\geq 10\%$), which is what was found in the error split analysis in section 4.3.3.5 on all datasets as shown in figures 4.14 and 4.15. However, to overcome this problem they have had to create a new dataset which costs in either money and/or time, which is not the case in the error split approach shown here. The approach of creating new datasets to examine properties of TDSA models is not scalable without large resources thus the error split approach is a very feasible alternative and as shown effective. Furthermore, using the new *STAC-Multi* metric it is now possible to quantify TDSA model performances on samples that only TDSA models can correctly predict. This does not mean that these challenge datasets are not useful, for instance using this dataset can help answer the question of whether using more DS_2 and DS_3 samples will improve TDSA models performance on those subsets and how much would that affect the performance on the DS_1 subset.

The baseline experiments (section 4.3.3) have brought about many different findings, of which some have confirmed prior findings, where as others have not. Unlike previous work it has been shown that TDSA models on datasets that do not contain a lot of targets per text such as the Laptop and Restaurant datasets can be statistically no better than a text classifier, prior work has shown this in absolute performance but not statistically (Jiang et al., 2019). From these baseline experiments we can conclude that all the error splits have been successfully tested across a range of models and datasets. From analysing

the error split results the baseline TDSA models generally perform best on subsets of data that contain one unique sentiment (DS_1) and on targets that appear multiple times in different sentiment classes within the training data ($KSKT$). This finding suggests that the baseline models are very brittle and cannot generalise to unknown targets (UT), unknown sentiment relations ($USKT$) or texts that contain multiple unique sentiments (DS_2 and DS_3). Due to these factors the models are unlikely to perform well in low resourced or cross domain settings. Subsection 4.3.4 introduced a novel TDSA metric *STAC* and its two variants *STAC-Multi* and *STAC 1* of which when used together can show sentiment overfitting to the most frequent sentiment class in a text to some extent. Furthermore the *STAC-Multi* shows how well the TDSA models can perform target sentiment relationship modelling perfectly, as well as how the *DS* subsets are a fine grained and easier version of *STAC Multi*.

Subsection 4.3.3.7 has related back to the original work that created these error splits in doing so explained why the *NT* split does not have a consistent trend. From exploring the different splits many of them have been dismissed due to the results not matching the hypothesis of what the split is supposed to measure. Thus the *NT* split, due to having no consistent trend, cannot be used to measure target interaction. The *TSSR* split cannot measure sentiment overfitting, and the *n-shot* split is not as useful as the *TSR*. Therefore the recommended splits to use are the *DS* for measuring fine grained target sentiment relation modelling and *TSR* to measure the model's ability to generalise to unseen targets and sentiment relationships. Furthermore, the *STAC-Multi* metric is recommended to measure target sentiment relation modelling, but it is a much stricter and coarser measure compared to *DS*. The *STAC 1* should also be used so that it can show to some degree with *STAC-Multi* the extent of overfitting to the most frequent sentiment class in a sentence.

Lastly, from exploring the results across these different splits future research directions have surfaced. Due to none of the splits being capable of explicitly measuring target interaction, an annotated corpus incorporating this annotation would be of use. He et al. (2018a) has shown that using an un-supervised autoencoder objective to mimic encoding a latent aspect into the target representations improves general results as well as results on multi word targets, and visually has shown on selected targets to create better target representations. However it would be of interest to see if such a method can help target representations for unknown targets (UT) as this would be similar to the Multi-Entity Aspect Based Sentiment Analysis (ME-ABSA) task, where Yang et al. (2018) had found no difference between UT and Known Sentiment Known Targets ($KSKT$). Thus suggesting that encoding the latent aspect could greatly benefit the UT samples.

4.5 Conclusion

The research question that this chapter was attempting to answer is RQ 3 'What is an appropriate empirical evaluation methodology for TDSA?'. To investigate this, section 4.2 reviewed the prior work in error analysis splits within TDSA. From this literature review, several existing error splits were found, *DS* which measured target sentiment relationship modelling, *NT* measuring target interaction, and *n-shot* measuring generalisation to unknown targets. From this literature review, two novel error splits were created, *TSSR* that measured target sentiment overfitting to the most frequent sentiment in a sentence

and *TSR* measuring generalisation to unknown sentiment relationships and targets. These existing error splits were rigorously tested in section 4.3.3 across three TDSA methods and a text classification method to ensure they were measuring what was hypothesised. From this, *NT* error split was removed due to it not measuring target interaction but rather the sentiment factors *DS*. *TSSR* was dropped due to it not measuring target sentiment overfitting without a text classification model and the *DS* split. Lastly, the *n-shot* split was removed as when the value of n increased it was expected the accuracy should increase as well or at the least not drop, which was not always true. Thus the *TSR* split which measured both unknown targets and sentiment relationships was recommended as a better replacement to *n-shot*. The findings from reviewing the *NT* split bring the recommendation that the only way to investigate target interaction is through an annotated corpus with this explicitly annotated. A novel TDSA metric is created, *STAC Multi* and *STAC 1*, which when used together can be used to evaluate sentiment overfitting to the most frequent sentiment in the sentence. Furthermore, the *STAC Multi* metric can be seen as a coarse grained version of the *DS* error split as they both measure target sentiment relationship modelling, but *STAC Multi* cannot be influenced by sentiment overfitting to the most frequent sentiment in the sentence.

In this chapter the error splits have been reduced to those that match their hypotheses (*DS* and *TSR*) and a new novel metric has been created to overcome previous limitations in the error splits. Therefore a new empirical evaluation methodology for TDSA has been created, whereby each error split and metric can be used to quantify different theories about a TDSA method.

Chapter 5

Case Studies in Improving Experimental Methodology for TDSA

5.1 Introduction[1]

Following on from chapter 4, several case studies will be explored using the newly developed TDSA evaluation methodology to test if it can quantify the justification behind these new developments. These justifications are normally qualitative case studies which are hard to quantify and to a large extent impossible to compare. Thus the importance on testing if this new TDSA evaluation methodology can work in practice is highly motivated to overcome the issues with qualitative analysis. Furthermore in each case study a rigorous experimental setup[2] will be conducted unlike many previous works.

The new developments, where each is a separate case study within the chapter, within the TDSA literature that will be tested are; encoding the target's position (position encoding) (Gu et al., 2018), inter-target encoding where each target is aware of all targets within the same text (Hazarika et al., 2018), and CWR (Sun et al., 2019a; Xu et al., 2019). Of these developments, only the first two are TDSA specific whereas the transfer learning is a general machine learning concept that has been shown useful in many NLP tasks (Peters et al., 2018a). These developments have been mainly justified by the improvements on the overall accuracy and/or macro F1 score, but the justification in the paper is normally more detailed. An example justification (and one that is typical of most developments) of inter-target encoding from Hazarika et al. (2018), where they first show improvements on the general accuracy scores over baselines. They then further state these improvements are due to the model being able to infer one target's sentiment from knowing another target's sentiment, and show this through a case study (section 4.2) from a few samples. The reason for inter-target encoding does sound valid but the case studies are qualitative and thus hard to quantify and compare too. Furthermore in

[1] All code that creates the evidence for this chapter can be found here: `https://github.com/apmoore1/tdsa_comparisons`. Certain sections throughout this chapter may have more specific pointers to python notebooks that created the analyses within that given section.

[2] This is through statistical testing, comparing across more models and datasets, and in the inter-target encoding setup comparing to a more suitable baseline model.

both position and inter-target encoding there have been papers by He et al. (2018a) and Majumder et al. (2018) that respectively use more detailed quantitative metrics through their own error splits to justify these improvements. However the error splits used in both cases (*NT*) have been shown within chapter 4 to be unsuitable.

All of these new developments will be applied to the methods that were used within chapter 4, as these methods can easily be enhanced with these new developments. Furthermore the same experimental setup that was used within chapter 4, described in section 4.3.2, will be used throughout this chapter.

5.2 Position Encoding[3]

5.2.1 Introduction

This is the first model enhancement that will be explored in this chapter. As stated in section 4.3.1, the two models that will be explored in this section are *IAN* and *Att-AE* due to the *TDLSTM* already having position information somewhat encoded into its NN architecture. Within the prior work, position information has been encoded into different TDSA methods in broadly three different approaches; weighting, embedding, and via the construction of the NN architecture (construction).

Position weighting is probably the simplest approach as it weights the vectors of tokens/words[4] based on some distance metric to the relevant target word(s). However there is no one standard distance metric in the literature but a lot of them are very similar; Chen et al. (2017) (equation 7 and section 3.3) based the weighting on how many tokens are between the context word and the nearest target word (token) and then normalised via sentence length[5]. Other methods have created an arbitrary cut off so that context words that are too distant are ignored (He et al., 2018a; Zhao et al., 2019), Zhang et al. (2019a) uses the same weighting as Chen et al. (2017) but ignores the target words. He et al. (2018a) incorporated syntax into the weighting where the position to the target word(s) is defined by the distance through the dependency tree. Lastly Li et al. (2018d) used the same weighting as Chen et al. (2017) but normalises using an arbitrary constant rather than the sentence length (n). In most cases across all of the experiments within the prior work on position weighting when the work has shown ablation studies position weighting has increased the performance of the models[6].

Position embeddings unlike the weighting mechanism encodes the position of a token/word via a learnt embedding space. Position embeddings are similar to the weighting mechanism in that they create position indexes that are relative to the target, where the indexes are created similar to the weighting mechanism. These position indexes for each word are calculated based on token distance from the closest target word. These position indexes, unlike the weighting method, are integers not floats as these integers are then used as an index to the random initialised (position) embedding. These position embeddings are then normally concatenated onto the word embeddings that represent

[3]All graphs within this section have been generated through the following notebook `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/Position_Encoding.ipynb`.

[4]Normally after they have been encoded via some sequence encoder e.g. LSTM.

[5]They further encoded the relative position of the word into the model.

[6]The only work that has shown position weighting to harm performance is Zhang et al. (2019a) on the Twitter and Rest14 datasets in table 3.

the tokens in the sentence that the target occurs in (Gu et al., 2018; Li et al., 2018b; Chen et al., 2019; Sun et al., 2019c; Kumar et al., 2020). Unlike the position weighting literature all prior works that have used position embeddings calculate the position indexes based on relative distance to the nearest target token. Out of the five prior works, three use a position embedding dimension of 100, one uses 50, and another 30[7]. For clarification on position weighting and embeddings, table 5.1 shows an example text that has been tokenised where the target that is being predicted for is 'Apple Mac', beneath each token is the weighting value and position index where the weighting value is calculated using equation 5.1, which has come from Chen et al. (2017). In equation 5.1, i represents the index of the token/word whose position is to be weighted, τ represents the index of the closest target token to i, and n is the length of the text in tokens.

$$w_i = 1 - \frac{|\tau - i|}{n} \tag{5.1}$$

Tokens:	The	Apple	Mac	is	great
Position indexes	1	0	0	1	2
Position weighting	0.8	1	1	0.8	0.6

Table 5.1: Example text which contains the target 'Apple Mac', where the text has been tokenised and the associated position indexes and weightings are shown.

Construction based approaches have used numerous different NN methods. Tang et al. (2016b) (*TDLSTM*) used RNNs where a forward RNN would process all tokens up to the last target token and a backward RNN for all tokens to the first target token. Other approaches have split the sentence up into left, right, and target contexts and aggregated the word embeddings using Neural Pooling methods (Vo et al., 2015; Zhang et al., 2016) or using a RNN based sequence encoder (Liu et al., 2017). Another direction is to make use of a dependency parser to explicitly model the syntactic structure of the target's position, Dong et al. (2014) re-ordered the dependency tree to ensure the target word is the root and then used a Recursive NN (RCNN). Sun et al. (2019c) and Huang et al. (2019) both use the dependency tree without re-ordering it and apply a Graph NN (GNN) to encode words that are close to the target word through the dependency tree. Nguyen et al. (2015) used both a dependency and constituency parser to create a phrase dependency tree where the the tree is re-ordered such that the target word phrase is always at the root of the tree, they then used a RCNN to encode the tree data. Lastly one prior work has combined both position embeddings and a position based architecture (Sun et al., 2019c).

From this prior literature, the main reason for using position information is to create a more explicit bias in the model. The bias assumes that words that are closer to the target word(s) are more important and therefore should be given more weight, attention, or priority in the model. This bias is more or less explicit depending on the method used, in the case of weighting it is more explicit as words that are closer are always weighted higher than those further away. In comparison the embedding approach can be less biased as it allows the model to choose which words are closer and more important through the embedding. The construction method is hard to compare to the embedding

[7]Another prior work does not state the dimension of the position embeddings (Du et al., 2019).

and weighting approaches on bias as these approaches to some degree change their whole model so that position information is prioritised. In the case of *TDLSTM* (Tang et al., 2016b) the bias of relatively close words are prioritised, compared to Dong et al. (2014) where they prioritise the syntactic distance between tokens and the targets words.

In general, encoding position information is used to improve the performance on sentences that contain multiple targets as it should help match relevant words with their respective target (Li et al., 2018b; He et al., 2018a). This indicates that the point of position encoding is to help improve target sentiment relation modelling. From these position encoding papers, He et al. (2018a) is the only one to quantitatively evaluate the importance of position information further than just using overall metrics on the entire dataset. He et al. (2018a) have shown that using syntactic position weighting improves the performance for sentences that contain more than one target. The way this was evaluated was in effect using the *NT* splits where they subsetted the data based on 1, 2, 3, and more than 3 targets per text. As has been shown in subsection 4.3.3, the performance on the *NT* split tends to be dominated by sentiment factors and does not directly measure target sentiment relationships. Even though He et al. (2018a) has performed a good quantitative evaluation, this cannot determine if the position information is actually improving target sentiment relation modelling, due to He et al. (2018a) using the *NT* split in their evaluation. Therefore in this section position encoding will be evaluated across the two TDSA methods using the recommended error splits from the last subsection *DS* and *TSR* as well as using the TDSA specific metrics *STAC Multi* and *STAC 1*. In doing so, position encoding will be thoroughly evaluated for target sentiment relation modelling that it is claimed to do from the literature, as well as exploring any other positive or negative side effects. Furthermore, the *NT* subsets will also be included in the evaluation to demonstrate the point of why this split is not suitable for error analysis as it has been used in the prior work (He et al., 2018a).

Given this prior literature the position encoding that will be used in this section is position weighting. This was chosen as it requires no extra parameters unlike the position embeddings and does not require fundamentally changing the NN architecture. The distance metric proposed by Chen et al. (2017) as shown in equation 5.1 will be used as it does not require any arbitrary cut off or normalising parameter to be tuned (Zhao et al., 2019; Li et al., 2018d), nor does it mask any words (Zhang et al., 2019a), and finally does not require a dependency parser (He et al., 2018a). Furthermore, as this weighting method does not remove any words it allows the model to ignore the explicit bias of the position weighting in cases where the closest words are not always the most important e.g. in the case that the affecting opinion word(s) are a few syntactic hops away from the target. The position weighting will be applied to the context/sentence vectors after being encoded by the LSTM layer and before the attention is applied in both *IAN* and *Att-AE*.

5.2.2 Experiments

Figure 5.1 shows the overall scores of the position encoded *IAN* and *Att-AE* models. These scores are somewhat meaningless without comparing them to their respective non-position encoded baseline models.

Figure 5.2 shows the metric score difference between the position and the respective baseline models. As we can see from the results the overall trend is that, no matter

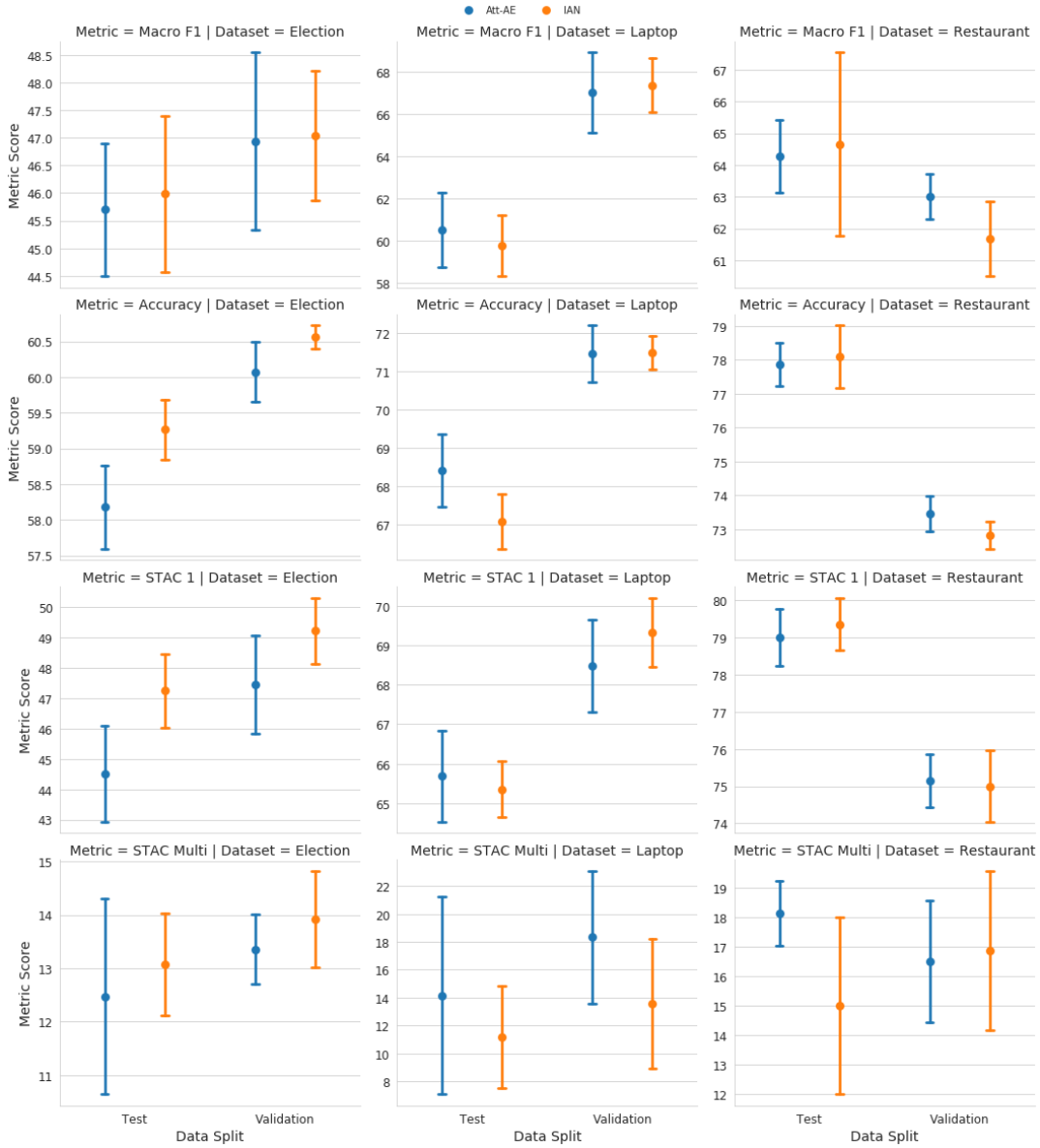


Figure 5.1: Columns represent different datasets, rows different metrics. Each plot represents the two position encoded models metric score on the test and validation splits.

the dataset or metric, position weighting on average improves the model's performance. There are a few exceptions to this in the *STAC 1* results for both Election and Laptop datasets, and the *accuracy* metric for the Laptop dataset. Furthermore there are several results where even though the mean is positive the standard deviations are so large that they go into the negative of the metric score difference. This shows that the trend might show an overall improvement in performance but the improvement is marginal.

To better visualise the differences figures 5.3 and 5.4 show the number of models that are significantly better than their baseline, where the former is not corrected for multiple significance tests whereas the latter is using Bonferroni and is aggregated across datasets.

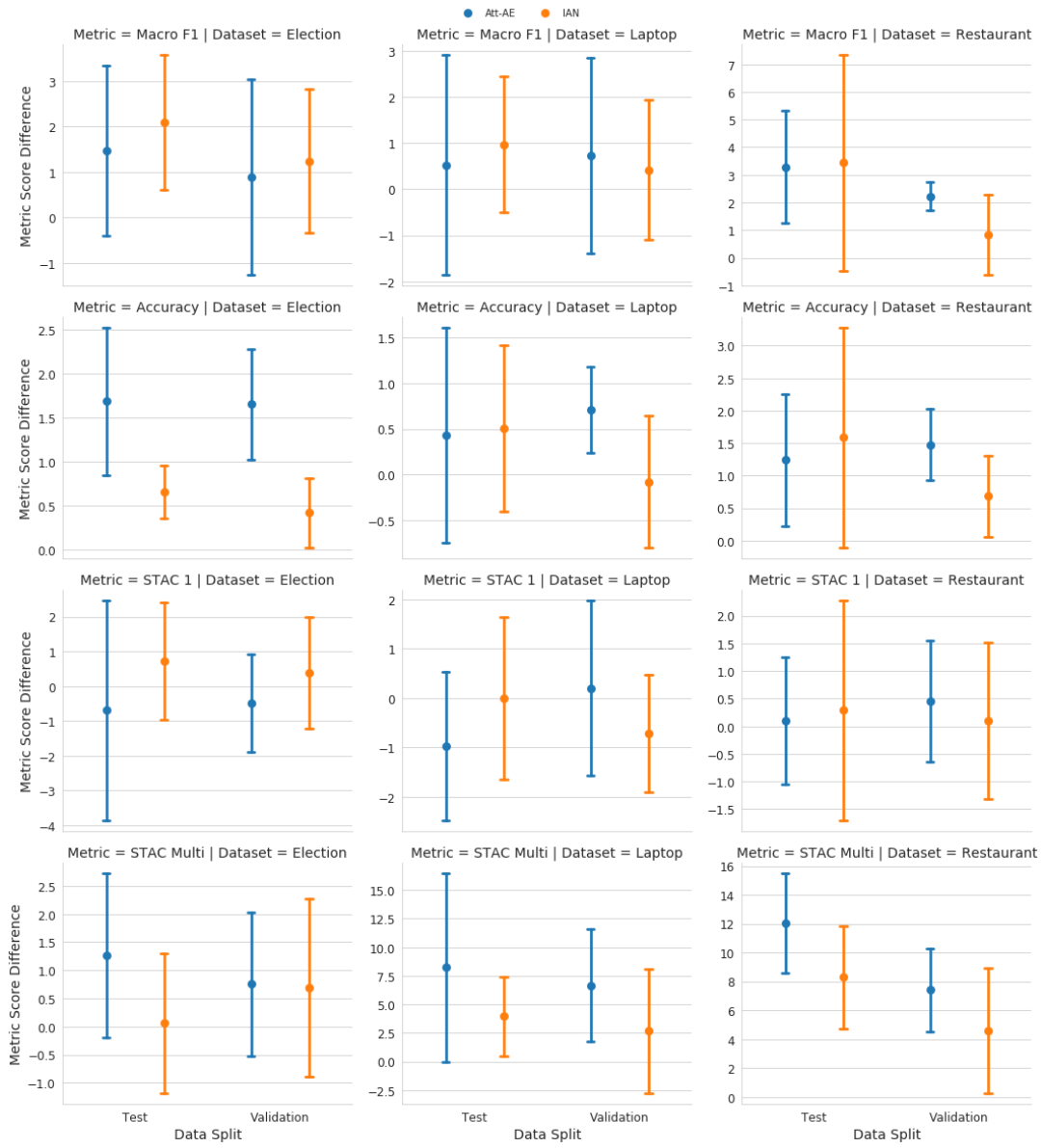


Figure 5.2: Columns represent different datasets, rows different metrics. Each plot represents the differences between the position and baseline models for the relevant metric score on the test and validation splits.

There were no models on any of the metrics or dataset splits where the baseline models were significantly better than the position models. From these heatmaps it can indeed be seen that the Restaurant dataset does benefit the most.

As can be seen from the heatmaps, the Laptop and the Restaurant datasets are the only datasets that consistently have position models that are significantly better on the *STAC Multi* metric. Furthermore, the Laptop dataset is the only dataset that does not find the position models to be consistently significantly better than the baselines on *Accuracy*. This might not be a surprising finding considering that the Laptop dataset

contains the least number of DS_2 and DS_3 samples relative to its size (see figure 4.1). Thus, the findings suggest that the position information improves the scores on the samples that have multiple unique sentiments within the sentences (DS_2 and DS_3), compared to a dataset that is made up of mainly sentences that only contain one unique sentiment. This reason is also related to another, of why position information does not perform significantly better on the *STAC 1* metric. As the *STAC 1* metric is only looking at the accuracy score for sentences that contain one unique sentiment and that the model predicts all samples in those sentences correctly. A model that overfits to the most frequent sentiment class will perform well on this metric. As the position information is biasing the model towards only looking at local context around the target it removes the bias for the model to look at the global context. This shows that biasing the model towards the local context through the position information removes some of the position overfitting that the baseline models exploited. It is clear to some degree that for at least the Laptop and Restaurant datasets that as the position models have improved on the *STAC Multi* metric and not regressed on the *STAC 1* metric, that the models are less prone to sentiment overfitting and do perform better at target sentiment relation modelling.

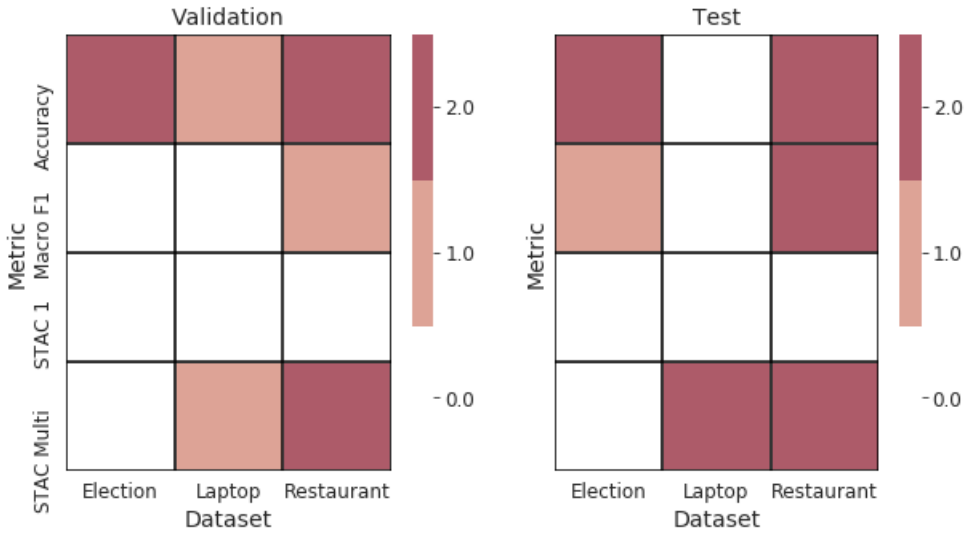


Figure 5.3: Heatmaps that represent the number of position models that are statistically significantly better than their baseline equivalents at the 95% confidence level.

Furthermore, from the heatmap results we can see that the Election dataset is the only one that the models do not perform significantly better than the baseline version on the *STAC Multi* metric. However, they do consistently perform better on the *Accuracy* metric. This may suggest that the position information does improve the target sentiment relationship modelling but cannot be shown through the *STAC Multi* metric as it requires all targets in the sentence to be correctly classified. The further reason why it is believed that it is improving the target sentiment relationship modelling rather than overfitting to the overall sentiment as the *STAC 1* metric has not improved on the dataset. To investigate if the position models are improving the target sentiment relationship modelling on the Election dataset the results across all datasets for the *DS*,

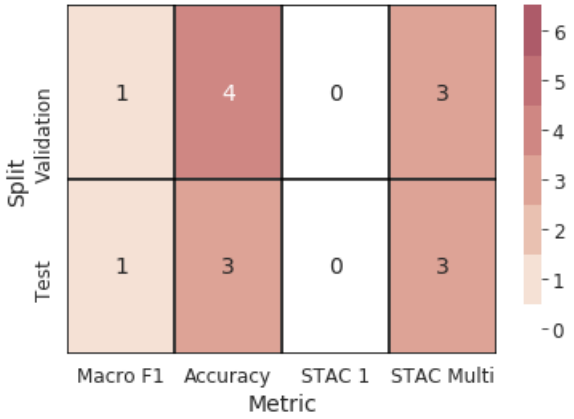


Figure 5.4: Heatmaps that represent the number of position models that are statistically significantly better than their baseline equivalents across all datasets at the 95% confidence level. Where the multiple hypothesis tests have been corrected using Bonferroni.

TSR, and *NT* error splits are shown in figures 5.5 and C.1 for the test and validation splits respectively. The figures show the difference between the position and respective baseline model[8]. From figure 5.5 it is clear that that the position models improve the results for the DS_2 subset on the Restaurant and Laptop datasets. However, for the Election dataset it would appear that the *Att-AE* model is the one that benefits most from the position encoding (better seen in the validation results). The heatmaps in figures 5.6 and 5.7[9] show the number of models that are significantly better than their baseline on each error subset, where the former is not corrected for multiple significance tests whereas the latter is using Bonferroni and is aggregated across datasets. The heatmaps differ by a large margin between the validation and test splits. The Laptop dataset from the validation split appears to be the only one that has no significant difference in any of the subsets, of which this differs from the overall metric results in figure 5.3 showing that there is a significant difference for the *STAC Multi* and *Accuracy*. It can be seen for the Election dataset for both validation and test splits that for all subsets in the *DS* split that a position model is significantly better. Furthermore, when correcting for multiple hypothesis tests on the test split figure 5.7 shows that the Election dataset must contain at least one position model that is significantly better for the DS_2 and DS_3 subsets. This to some degree confirms that the position encoding must be improving the target sentiment relationship modelling even though it does not show through the *STAC Multi* metric. This shows the reason why the *DS* split is still useful as it shows a more fine grained analysis of the target sentiment relationship modelling.

As was suggested in sub section 4.3.3 the *DS* and *TSR* splits should be used when analysing the results, this is the reason why the *TSR* split was included in the figures. From figure 5.7 for the *TSR* split it can be seen that in general the only subset that the position models consistently perform better on is the *KSKT* subset. This is most likely

[8]Figures C.2 and C.3 show the test and validation results respectively for the splits without subtracting from their respective baseline models.

[9]The legend goes to 6 as the number of position models that can be better than their baseline is 6, as there are 2 models and 3 datasets per evaluation, and each model is evaluated against its baseline across all datasets.

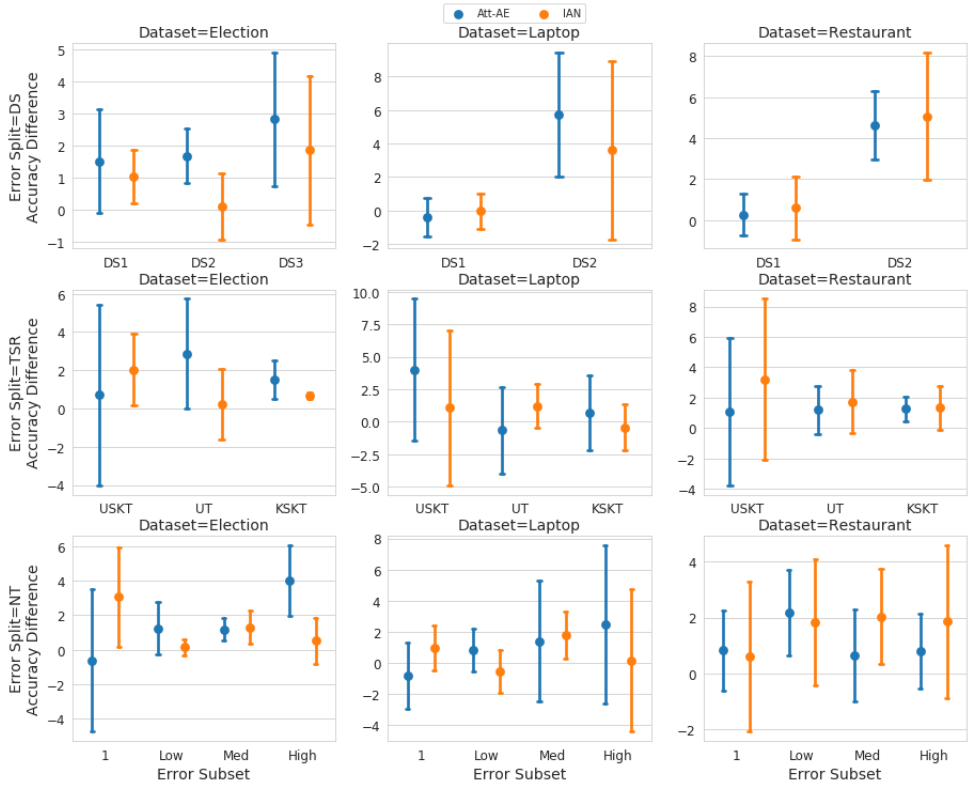


Figure 5.5: Test split results. Columns represent different datasets, rows different error splits. Each plot represents the differences between the position and baseline models for the Accuracy metric on the given error subset.

due to the target sentiment relationship modelling improvement, as the *KSKT* subset is the dominate subset within the *TSR* split (see figure 4.11).

Lastly, as in the previous work by He et al. (2018a), the way they evaluated the position encoding was through using a similar method to the *NT* error splits. Therefore to show quantitatively that the *NT* splits are not the best way to show that a method improves on target sentiment relationship modelling, figure 5.7 finds that only a few models are significantly better across the different *NT* subsets. When any of the *NT* subsets are compared to the DS_2 subset none have more significant models. Furthermore, figures 5.8 and 5.9 show the validation and test split performance difference between the position and baseline models when the data has been first subsetted by the *DS* subsets (rows in the figures) and then further subsetted by the *NT* subsets. From these figures it shows some inconsistencies of which the most of these are in the validation results. The validation results for the Restaurant dataset when subsetted by DS_1 show a large negative correlation of which this is most likely due to the baseline model being better at sentiment overfitting on sentences that contain lots of targets. Furthermore, on the Laptop test and validation splits for the DS_1 subsetted row several of the results mean value is less than 0 suggesting the baseline model is better in these circumstances. From across the test and validation splits out of all the DS_1 subsetted models 29.16%($\frac{14}{48}$) perform on average worse than their baseline models compared to 10.4%($\frac{5}{48}$) from the

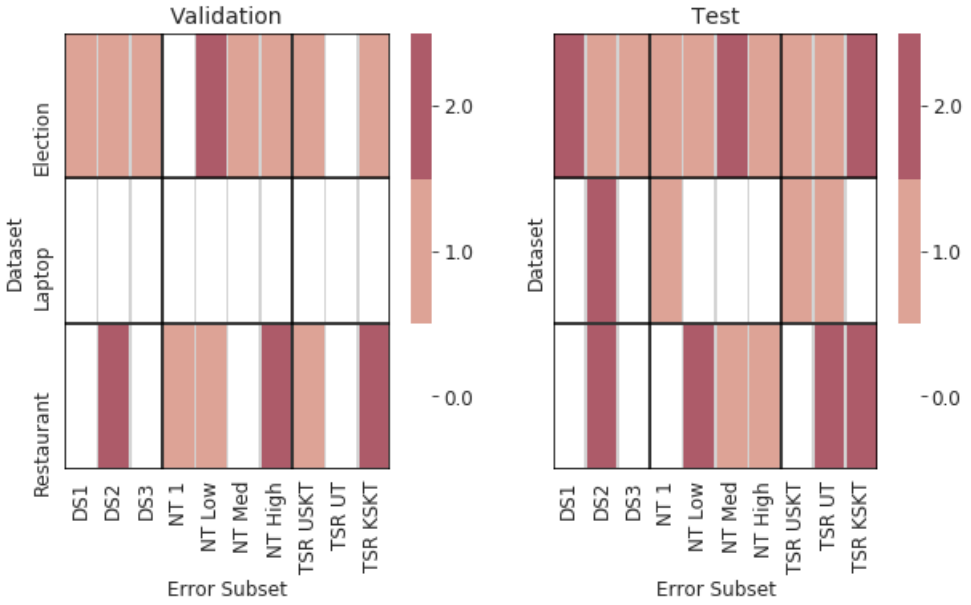


Figure 5.6: Heatmaps that represent the number of position models that are statistically significantly better than their baseline equivalents at the 95% confidence level based on the accuracy metric.

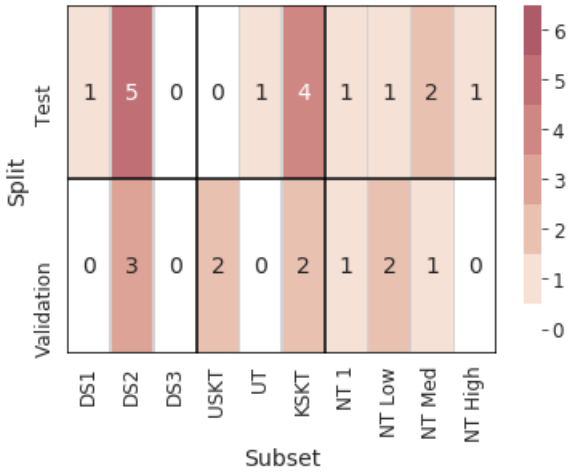


Figure 5.7: Heatmaps that represent the number of position models that are statistically significantly better than their baseline equivalents across all datasets at the 95% confidence level based on the accuracy metric. Where the multiple hypothesis tests have been corrected using Bonferroni.

DS_2 and DS_3[10] subsetted models. This to some degree shows that the NT split for the DS_1 subsetted data at least is more likely measuring sentiment overfitting than target sentiment relationship modelling, as the baseline models perform competitively. From

[10]The DS_3 subsetted models from the Laptop and Restaurant datasets were not included as they contain very few samples (less than 20).

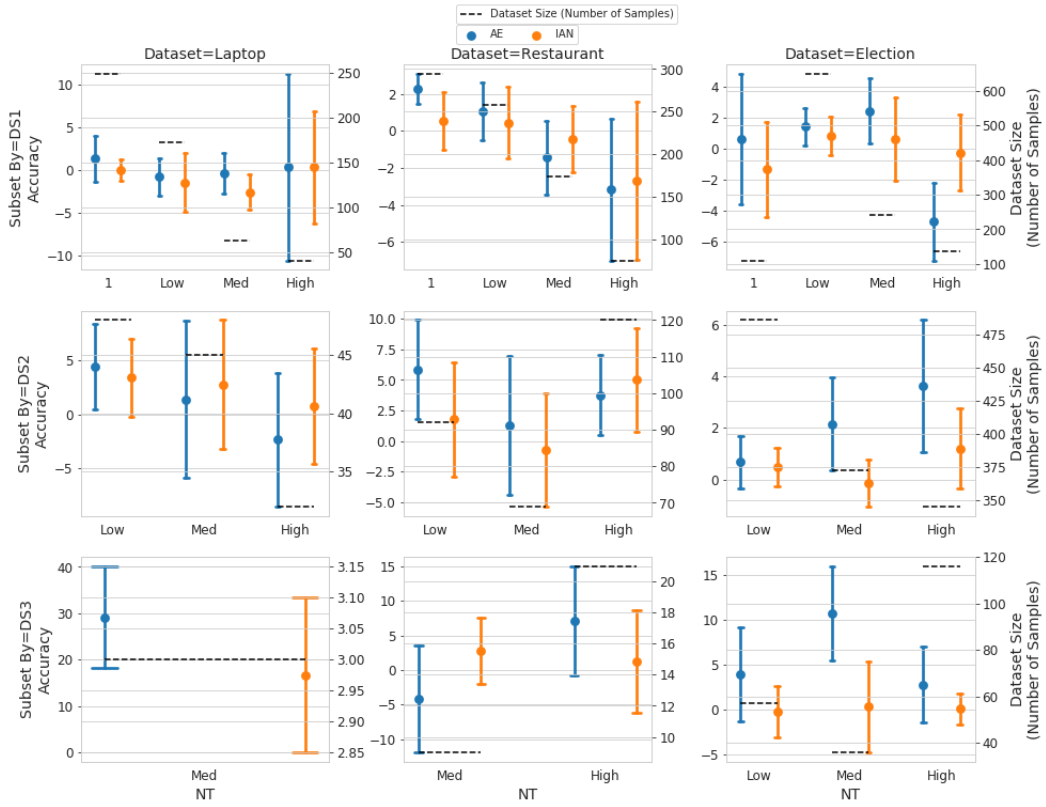


Figure 5.8: Validation split results. Columns represent different datasets, rows different *DS* error split subsets. Each plot represents the differences between the position and baseline models for the Accuracy metric on the given error subset.

this it can be seen that when the number of targets does increase the performance of the position models do not always improve the results. Furthermore, the *NT* split is more affected by the sentiment factors within the sentence and can be skewed by sentiment overfitting.

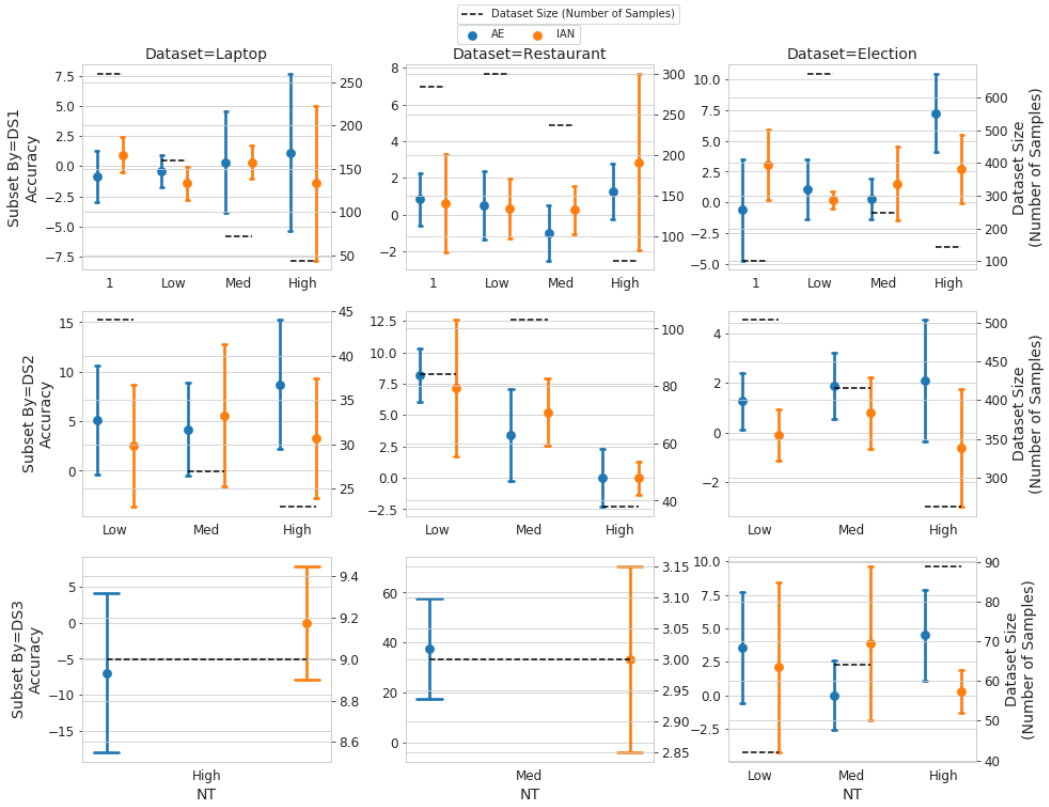


Figure 5.9: Test split results. Columns represent different datasets, rows different *DS* error split subsets. Each plot represents the differences between the position and baseline models for the Accuracy metric on the given error subset.

5.2.3 Conclusion

This section has found that position encoding does improve TDSA models performance in general. Further it has been shown quantitatively that the theory from the literature on position encoding improving target sentiment relationship modelling (Li et al., 2018b; He et al., 2018a) is true through the novel TDSA metrics (*STAC Multi* and *STAC 1*) and existing error split (*DS*). This finding could explain the reason why the *TDLSTM* model performed generally better than the others on the *STAC Multi* metric, and DS_2 and DS_3 subsets within the baseline results section (4.3.3). Lastly, it has been shown again that the *NT* split is not a useful error split to use due to it being skewed by sentiment overfitting. From this the results that He et al. (2018a) presented using a similar error analysis split as *NT* does not quantitatively prove that position encoding does improve target sentiment relationship modelling. Therefore, the results presented here are the first quantitative results demonstrating that position encoding does improve target sentiment relationship modelling.

5.3 Inter-Target Encoding[11]

5.3.1 Introduction

The majority of TDSA methods do not explicitly take into account other targets that may occur within the same sentence as the target that it is currently classifying, methods that do take this into account are target-aware. The standard non-target aware methods generally treat the problem as shown in figure 5.10a, where for each target (from the three) within the sentence is inputted into the (TDSA) model individually to create a target sentiment aware representation for the target (red/pink square), this representation is then projected down to a vector $\mathbb{R}^c$, where c is the number of sentiment classes for prediction. Whereas a target-aware model as shown in figure 5.10b takes all of the targets as input and makes predictions on all targets (three of them in this case) in-effect at the same time, where the model explicitly makes itself aware of all targets within the text through the inter-target encoding layer. The inter-target encoding layer was first suggested by Hazarika et al. (2018), where they used an LSTM as their inter-target encoder as shown in figure 5.10c. As can be seen the LSTM layer is uni-directional thus all targets preceding other targets within the sentence would not be aware of these future targets. This limitation was overcome by Majumder et al. (2018) where they used a similar network to Hazarika et al. (2018) but added a memory network on top, of which this memory network performed attention so that each target sentiment representation was aware of all other targets. Zhao et al. (2019) instead of using attention and an RNN structure made the targets aware of each other explicitly through a GNN, as shown in figure 5.10d. This inter-target encoding layer method is one of two general ways of making a model target aware. The other approach is through regularisation of the attention layer (Fan et al., 2018) that is applied to the encoded word representations within the sentence, to make the representations target aware. This form of regularisation was called 'Aspect Alignment Loss'[12] (Fan et al., 2018), where the intuition behind it was to penalise similar attention weight of targets in the same sentence that contain different sentiment[13]. This was to try and force the attention weights of targets that have different sentiment to focus on different words within the sentence. The only other work that has used regularisation to make a model target aware is Hu et al. (2019a). Hu et al. (2019a) applied a very similar technique to Fan et al. (2018), however this work was done on the task of aspect based sentiment analysis rather than target[14].

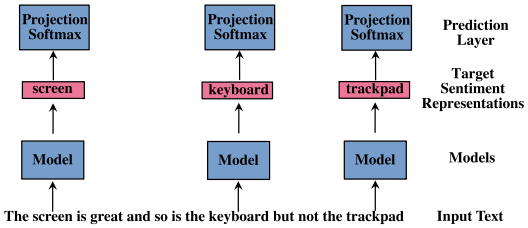
The main benefit that these papers believe that the inter target encoding layer brings, is the improvement for targets that depend on the sentiment of other targets within the same text. This target interaction was hypothesised within sub section 4.2.1 when describing the different error splits to be quantifiable through the NT split. However this was shown not to be the case through the experiments in section 4.3.3. Thus testing the theory of whether target aware models do perform better on targets that do rely on others cannot be tested here, this would require a specialist corpus as stated in chapter 4. Furthermore, only one prior work (Majumder et al., 2018) tested this hypothesis more

[11] All graphs within this section have been generated through the following notebook `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/Aspect_Encoding.ipynb`.

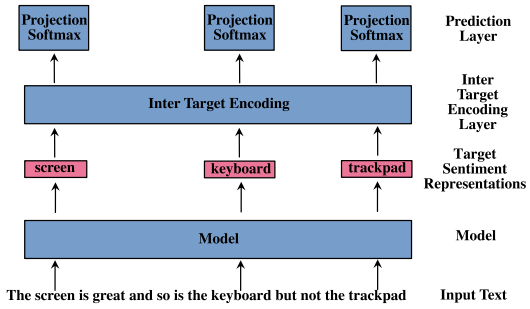
[12] Aspect here is the same as target.

[13] Equation 24 shows the Aspect Alignment Loss in Fan et al. (2018).

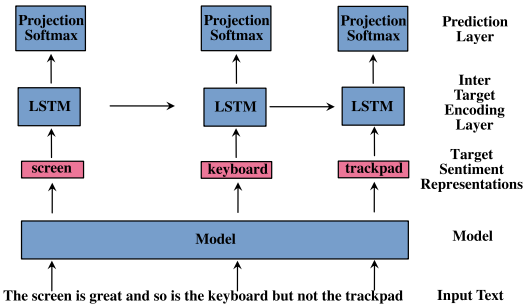
[14] Aspect based in this thesis is the latent version of target based.



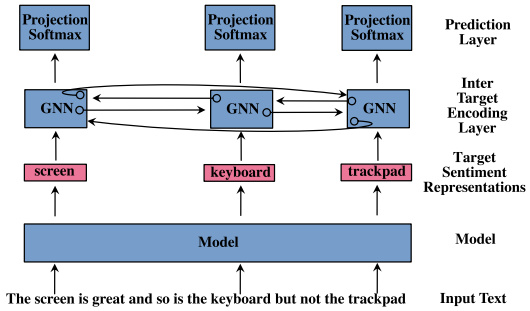
(a) General non-target aware setup.



(b) General target aware setup.



(c) Target aware setup of Hazarika et al. (2018).



(d) Target aware setup of Zhao et al. (2019).

Figure 5.10: The first row shows in abstract the difference between non target aware and aware setups. The second row shows more concrete target aware methods.

than just qualitatively showing some samples, or dataset statistics on the number of sentences that contain multiple targets. Majumder et al. (2018) stated the results of their target encoding method on two error subsets of the data, the subset that contains sentence with one target and the sentences that contain more than one target. These two subsets are equivalent to *NT 1-target* and the combination of all of the other *NT* subsets combined respectively. Even though this error analysis Majumder et al. (2018) performed would not directly test the target interaction hypothesis, it was an alternative form of measurement. This alternative form of measurement, as shown in section 4.3.3 would not measure target interaction. However, in these prior works including the regularisation work (Fan et al., 2018), it was suggested that target aware methods should perform better in sentences that contain multiple targets. Hence the error analysis in Majumder et al. (2018) was somewhat a valid choice. This form of coarse measuring of sentences that contain one target and others that contain multiple targets does not measure how good a method is on multiple targets. Rather this measurement is mainly dominated by sentiment factors as shown in section 4.3.3 where sentences that contain one sentiment are easier to classify when they have multiple targets rather than one, even for text classifier methods. Thus knowing that using the number of targets in a sentence cannot state anything about the method in general, the models here will not be evaluated using the same error split as Majumder et al. (2018). The models within this section will use the recommended error splits and metrics from section 4.3.3. These error splits and metrics will inform the community on what the target aware models actually capture more than their baseline equivalents.

The target aware method used in this section is the LSTM approach by Hazarika et al. (2018) due to the model that is used within the paper to create the target sentiment aware representations being that of the *Att-AE* method. Therefore the *Att-AE* model when enhanced within the inter target encoding will be the same as the model used in Hazarika et al. (2018). This will allow direct comparisons with the overall results from Hazarika et al. (2018). Furthermore, the approach from Hazarika et al. (2018) was chosen over Majumder et al. (2018) due to it containing far fewer components within the inter-target encoding and the results being only slightly worse[15]. As stated in the introduction the target aware enhancement will be added to all of the TDSA models.

5.3.2 Experiments

As stated in the introduction to this section the *Att-AE* model when enhanced with the LSTM target encoding of Hazarika et al. (2018) becomes the same model that was used in Hazarika et al. (2018). Therefore the first experimental results presented in figure 5.11 shows the single run performance from the reported accuracy scores of Hazarika et al. (2018)[16] and the distribution of eight scores from the reproduced version in this thesis. As can be seen the reproduced model does not contain the original model's stated performance within its distribution of scores, thus it failed to reproduce the original model. The reproduced model does use the same hyperparameters and components as those stated in Hazarika et al. (2018), the only parameter difference is the output dimension of

[15]Hazarika et al. (2018) (Majumder et al. (2018)) reported 79% (80%) and 72.5% (73.8%) accuracy on the Restaurant and Laptop datasets respectively. The reported results are only one accuracy score as they did not run the models multiple times to take into account the random seed/initialisation problem.

[16]The accuracy scores were taken from table 2.

the LSTM that creates a representation for the target words,[17] in the reproduced model it is 300 whereas in the original it is 100. The reason for it being 300 dimension instead of 100 is that the *Att-AE* model is based on two prior works Hazarika et al. (2018) and Wang et al. (2016b) and in Wang et al. (2016b) the parameter is 300 not 100[18]. Even though the *Att-AE* model could not reproduce that of Hazarika et al. (2018), it may not be due to the different parameter decision on the LSTM. As Hazarika et al. (2018) report similar models that use their inter target encoding layer but have results more similar to those of the reproduced model 74.5% and 73.42% on the Restaurant dataset and 69.6% and 63.7% of the Laptop dataset. As the results reported in Hazarika et al. (2018) are single run scores, it is hard to determine if the LSTM parameter difference is the reason or the authors had a favourable random seed, as it has been shown for the macro F1 score that NN TDSA models can range by up to 15 F1 points (Moss et al., 2019)[19]. Even though the results are not reproduced, it is still reasonable to compare and contrast the results from the reproduced models that use target encoding and the models that do not. As models being compared are reproductions rather than comparing to the original works results directly.

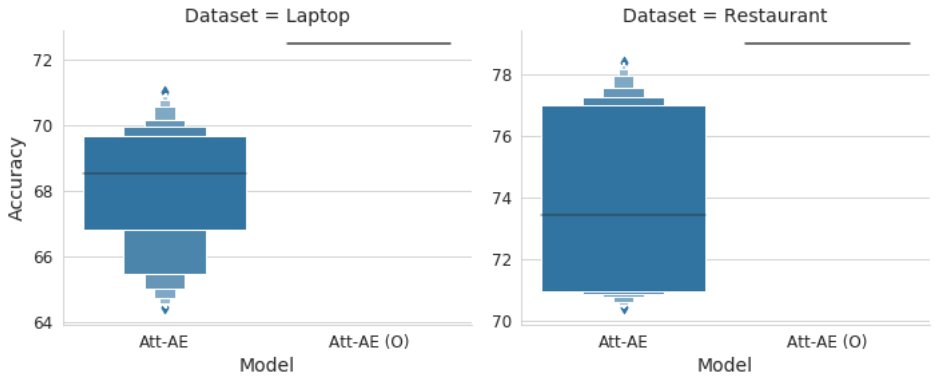


Figure 5.11: Distribution of eight scores and the line represents the mean value, for the original model this line represents their only reported score. Model name with a *(O)* represents the score reported in the original models paper.

Figure 5.12 presents the overall results across the different metrics for the inter target encoding models. For easier comparison, figure 5.13 reports the metric differences between the inter target encoding models and their respective baseline. As can be seen the results are either no better or worse in the majority of cases, only for the *TDLSTM* model for the *STAC 1* metric on the Election dataset are the results better as the standard deviation bars are greater than zero. This shows without performing any statistical tests that, in general, target aware models are not any better than their baseline models.

Thus the heatmaps in figures 5.14 and 5.15 show for the left and right column the number of target aware and baseline models respectively that are statistically significantly better than their respective baseline and target aware models, where the former (figure 5.14) is not corrected for multiple significance tests whereas the latter (figure 5.15) is using Bonferroni and is aggregated across datasets. As can be seen the only significant

[17]This is LSTM_a within section 3.1 of Hazarika et al. (2018).
[18]See the first paragraph of section 4 in Wang et al. (2016b).
[19]See figure 3.

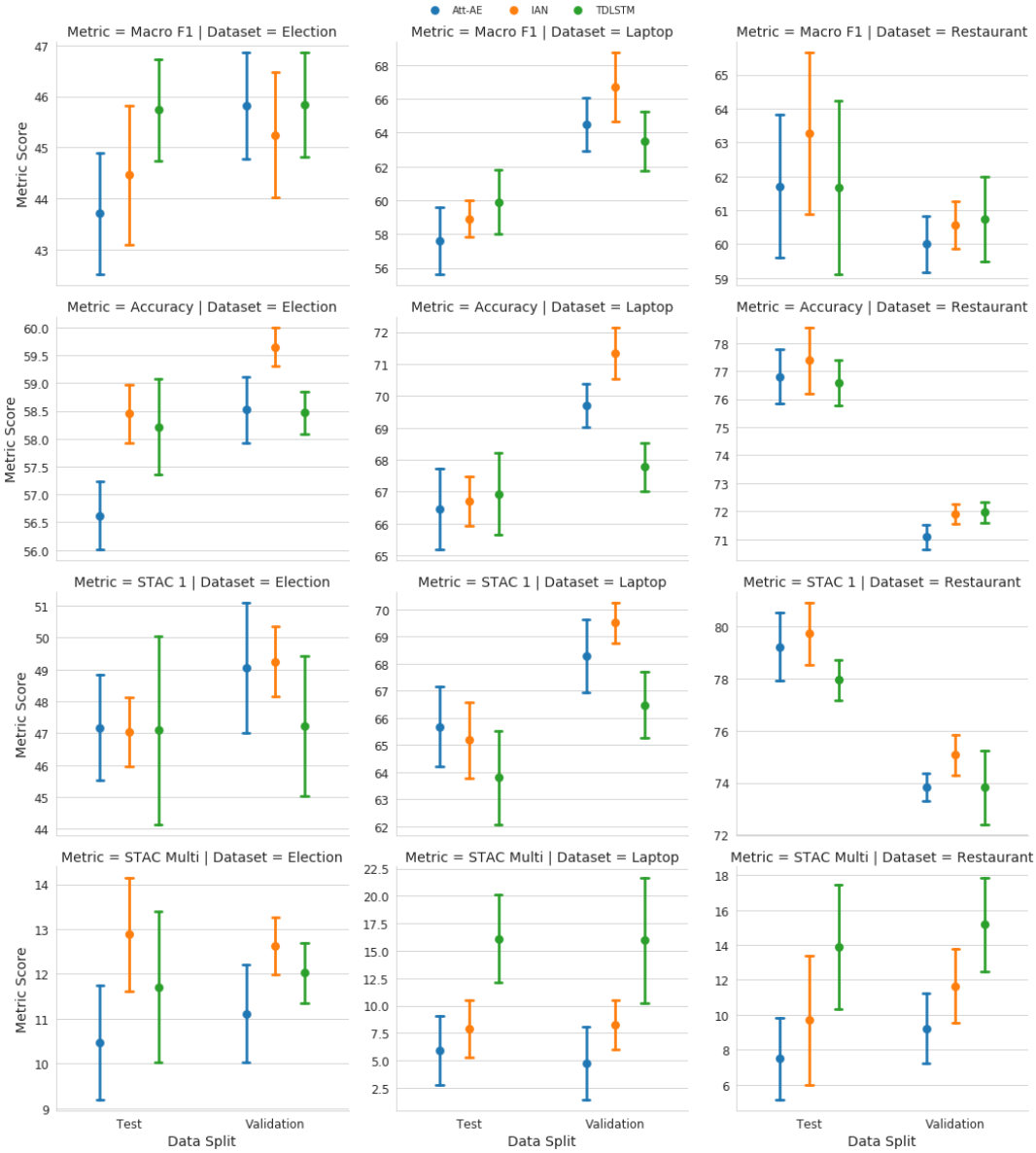


Figure 5.12: Each plot represents the three target aware enhanced models metric score on the test and validation splits. Columns represent different datasets, rows different metrics.

metric difference for the target aware models is on the *STAC 1* metric for the Election dataset, of which when corrected with Bonferroni does not exist. On the other hand the baseline models appear to be better on the accuracy and *STAC Multi* metric across both test and validation splits. This may suggest that for the Election dataset at least that the target aware models are somewhat overfitting to the most frequent sentiment more as they improve the results on the *STAC 1* metric while becoming worse on the *STAC Multi* metric. This would seem intuitive as the model could take advantage of knowing roughly the sentiment of all of the other targets within the sentence due to its target

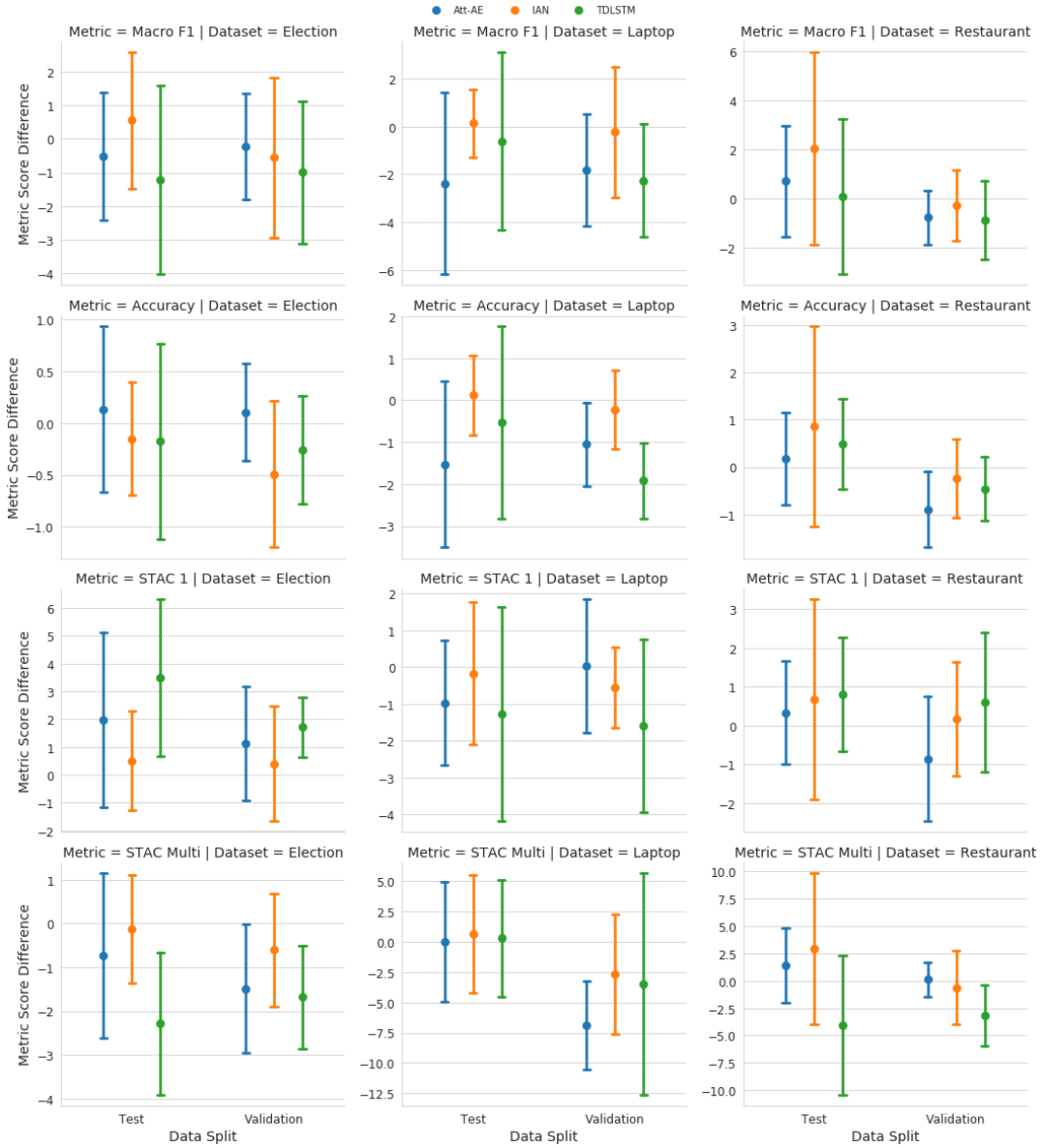


Figure 5.13: Each plot represents the differences between the target aware and the baseline models for the relevant metric score on the test and validation splits. Columns represent different datasets, rows different metrics.

awareness, and thus take the most likely overall sentiment. However, when taking into account multiple tests (figure 5.15), only two of the metrics show any differences between the target aware and their baseline equivalents, of which both of these differences only occur on two of the nine possible models both showing that the baseline models are better than their target aware equivalents. Thus showing the differences between the target aware and the baselines models are small, and that the baseline models are always at least as good as their target aware equivalent.

The results comparing target aware to the baseline models across *DS* and *TSR* error

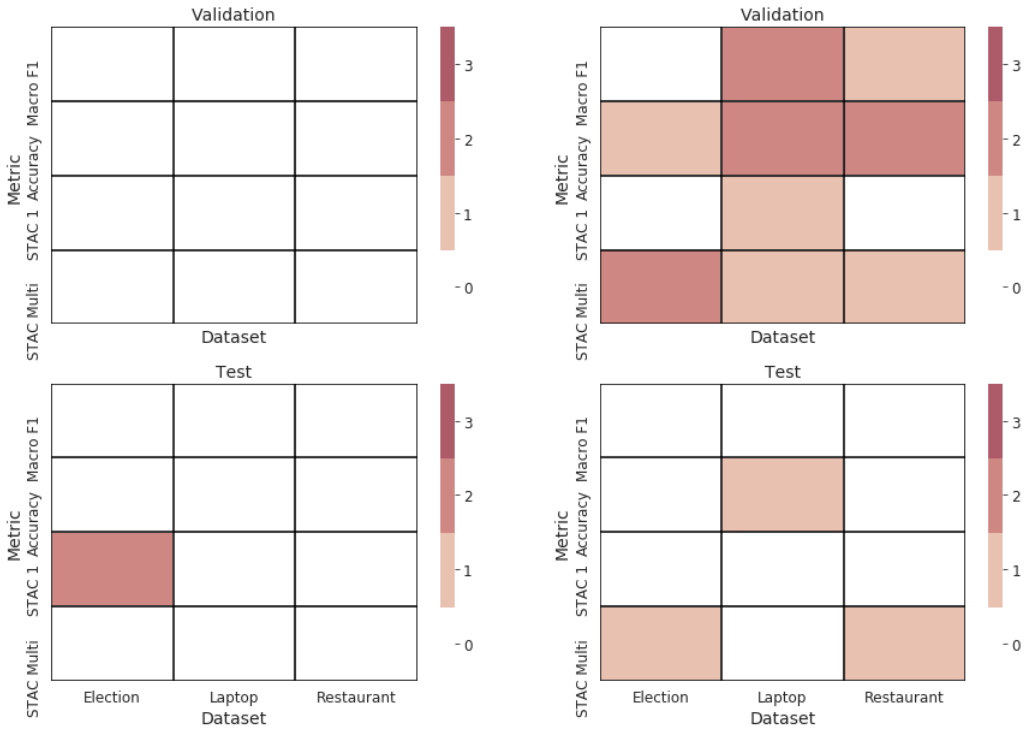


Figure 5.14: The left hand side heatmaps represent the number of target aware models that are statistically significantly better than their baseline equivalents at the 95% confidence level. The right hand side heatmaps represent the number of baseline models that are statistically significantly better than their target aware equivalents at the 95% confidence level.

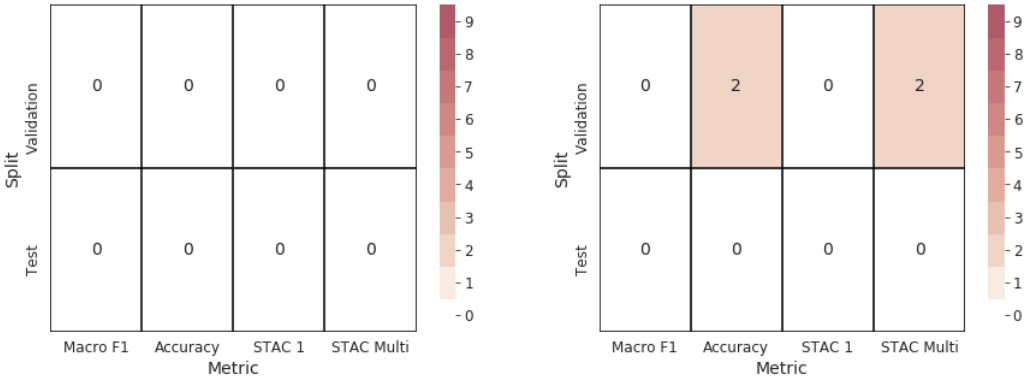


Figure 5.15: The left hand side heatmaps represent the number of target aware models that are statistically significantly better than their baseline equivalents across all datasets at the 95% confidence level. The right hand side heatmaps represent the number of baseline models that are statistically significantly better than their target aware equivalents across all datasets at the 95% confidence level. In both cases the multiple hypothesis tests have been corrected using Bonferroni.

splits are shown in figures 5.16 and 5.17 [20]. These heatmaps (figures 5.16 and 5.17) show once again that the target aware models are not any better than their baseline equivalents on any subset of any split of any dataset consistently across data splits. Furthermore, even though the baseline models are significantly better and consistently better for some subsets of splits as shown in figure 5.16 when corrected for multiple testing they are not, as shown in figure 5.17. Thus this shows once again the target aware models are no better if not worse in some cases than their baseline equivalents.

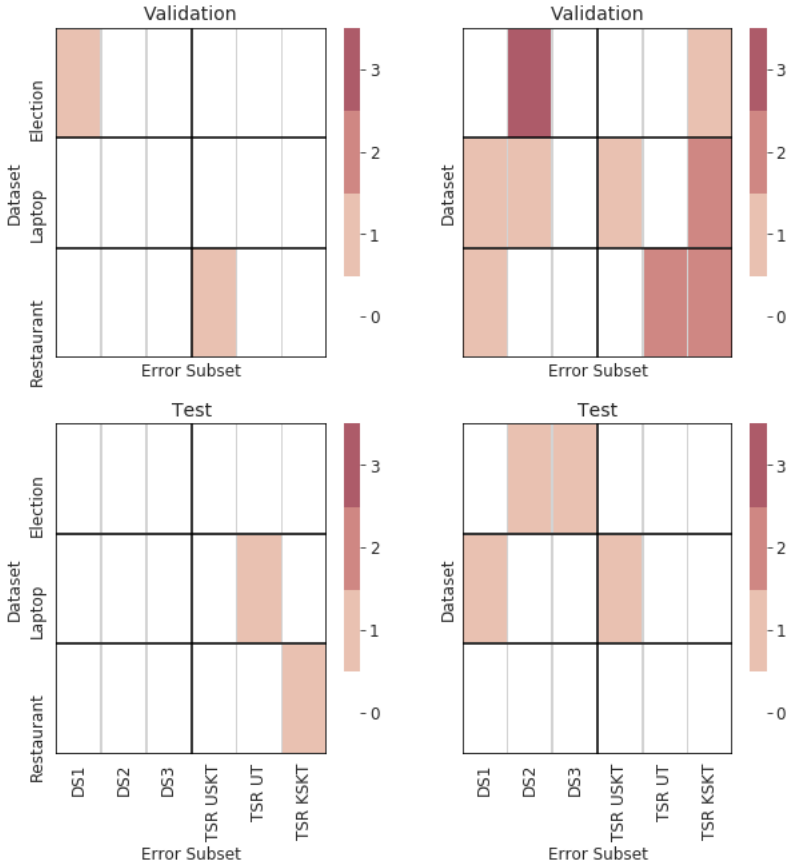


Figure 5.16: The left hand side heatmaps represent the number of target aware models that are statistically significantly better than their baseline equivalents at the 95% confidence level. The right hand side heatmaps represent the number of baseline models that are statistically significantly better than their target aware equivalents at the 95% confidence level. In both cases are based on the accuracy metric.

[20] The accuracy results for each subset for the test and validation splits are shown in figures C.4 and C.5 respectively. The accuracy difference between the target aware and their associated baseline models are in figures C.6 and C.7 for the test and validation split respectively. These results are within the appendix as they do not show any additional information that the heatmaps in figures do not represent better. They are also within the thesis itself for reproducibility reasons.

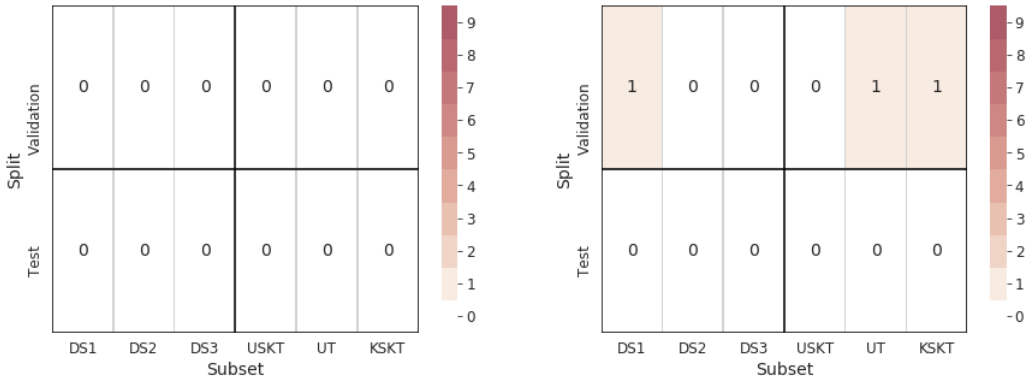


Figure 5.17: The left hand side heatmaps represent the number of target aware models that are statistically significantly better than their baseline equivalents across all datasets at the 95% confidence level. The right hand side heatmaps represent the number of baseline models that are statistically significantly better than their target aware equivalents across all datasets at the 95% confidence level. In both cases the multiple hypothesis tests have been corrected using Bonferroni and evaluated using the accuracy metric.

5.3.3 Conclusion

To conclude, even though the previous works that incorporated some form of target awareness into their models have shown improvements over their own baseline models (Zhao et al., 2019; Fan et al., 2018), this has not been shown here. Furthermore, it has been shown that on some datasets and some baseline models that they are significantly better than their target aware version. This negative result showing that unlike the previous work here it has been shown that the target aware models are no better than their baseline models, this could be due to none of the previous works performing rigorous testing of their methods. In all of the previous works, none perform statistical significance testing nor do they take into account the random seed problem (Reimers et al., 2017) which has been shown significant in chapter 3 for NN TDSA methods. In comparison this work takes both the significance testing and random seeds into account. Even though it was shown that the *Att-AE* model could not reproduce the results from the original paper (Hazarika et al., 2018), it can be concluded here that at least for Hazarika et al. (2018) inter-target encoding method it does not perform any better than not using it. Even though it is stating that opposite of Hazarika et al. (2018), the results here are more rigorously tested and on more models and datasets.

5.4 Contextualised Word Representations (CWR)[21]

5.4.1 Introduction

The majority of TDSA work so far has only used the standard 840 billion token 300 dimensional GloVe vectors (Pennington et al., 2014) to initialise word representation within the NN (Tang et al., 2016a; Tang et al., 2019). These non-CWR word vectors in

[21] All graphs within this section have been generated through the following notebook `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/CWR.ipynb`.

effect transfer semantic and syntactic knowledge of words from the semi-supervised task they were trained from (Mikolov et al., 2013b). Due to this transferring of knowledge they have been shown to help in multiple NLP settings such as Semantic Role Labelling (Collobert et al., 2008), Named Entity Recognition, Chunking (Turian et al., 2010), and text classification (Kim, 2014). The word vectors are normally the embedding layer (the first layer) of the NN that is usually trained with a Language Model (LM) type of objective. Thus the data they train from does not require any human annotation. Due to the word vectors coming from this embedding layer the vectors themselves are not contextualised as the NN at this point has not seen any of the other words within the context/text. Therefore, the main drawback of non-CWR is that they suffer from word ambiguity (Camacho-Collados et al., 2018).

To overcome this, CWR have been devised which are still trained on a LM objective, but instead of using the embedding layer they either use the second to last layer of the NN (Peters et al., 2017), or a weighted combination of each layers weights which has been shown to be more effective (Peters et al., 2018a). These CWR have been shown to outperform non-CWR across a spectrum of NLP tasks by a large margin (Liu et al., 2019a). The CWR work that has been stated are used in a *feature based* approach where the CWR are fed to a task specific architecture. However, work has also been focused on an alternative approach *fine-tuning* where normally the last layer of the LM NN is replaced with a new task specific layer, and then the whole NN is tuned to the new task (Radford et al., 2018; Howard et al., 2018). This approach has become very popular since the arrival of BERT (Devlin et al., 2019), due to its impressive performance across multiple tasks compared to other State Of The Art (SOTA) approaches, and its accessibility through the release of its code base[22].

For the interested reader on CWR, there are many different CWR each with subtle differences. These differences are normally based around their learning objective e.g. Bi-LM (Peters et al., 2018a), masked LM (Devlin et al., 2019), etc, how they are trained, e.g. discriminative fine-tuning (Howard et al., 2018), model design choices (Liu et al., 2019c), distillation (Tsai et al., 2019), etc if they are multi-lingual (Conneau et al., 2019), multi task learning with supervised objectives (Liu et al., 2019b), for a whole list of papers on CWR see the following GitHub list[23].

As stated in the introduction section 5.1, previous works have already started to use CWR within TDSA, of which a comprehensive list of these works and metadata details can be seen in table 5.2[24]. The table clearly shows that a lot of the prior works have tried different architectures: Task Specific Architecture (TSA) and non-TSA[25], fine-tuning or feature based, and pre-training the CWR and not. From this prior work some general insights can be drawn, which will be discussed below.

Pre-training in this thesis is defined as the process of fine-tuning the LM NN that the CWR come from with a separate task and dataset before using the LM NN on the

[22] `https://github.com/google-research/bert`

[23] `https://github.com/thunlp/PLMpapers`

[24] This is most likely already out of date due to the speed that new publications are coming out, and the fact that CWR produce the SOTA results. The raw data that has created this table can be found at the following URL: `https://github.com/apmoore1/tdsa_comparisons/blob/master/Overview_of_TDSA_methods_that_use_CWR.csv`.

[25] non-TSA is defined as adding a linear layer on top of the CWR class vector in the case of the BERT model (which is the only CWR model used in prior work).

final task, which here is TDSA. A concrete example of pre-training from table 5.2, is pre-training to Yelp and/or Amazon datasets by using an LM objective to fine-tune the BERT_b LM NN to these dataset, thus making the BERT_b model more domain specific. Generally one would expect pre-training to improve results as it adapts the CWR to the domain, and this is what both Rietzler et al. (2020) and Xu et al. (2019) found. However, considering both works use the same pre-training datasets and technique, as well as same CWR and model architecture there is a 2-3% difference in results. This is most likely due to the lack of pre-training that Xu et al. (2019)[26] did not perform, as Rietzler et al. (2020) showed that at least 10 million sentences[27] are required before any improvements can be seen for the Laptop domain.

Another insight that can be drawn from the works that use TSAs (Zeng et al., 2019; Zhao et al., 2019; Song et al., 2019; Huang et al., 2019; Jiang et al., 2019) is that using a CWR instead of a non-CWR is always better. This finding is more interesting for the works that do not fine-tune the CWR (Zhao et al., 2019; Huang et al., 2019) as it shows that by using CWR alone, and not more parameters, results improve over using non-CWR. Furthermore Song et al. (2019), Jiang et al. (2019), and Huang et al. (2019) all found that using a TSA with CWR in a feature based or fine-tuning approach to be better than fine tuning the CWR model with no TSA.

Lastly shown in table 5.3 are the best performing non-CWR methods[28], of which these are a lot worse in performance compared to any of the CWR methods within table 5.2.

[26] Used around 1 and 2 million sentences for Laptop and Restaurant domain respectively. This was calculated by multiplying the batch size (16) by the number of training steps.

[27] These do not have to be the same sentences, as they had 1 million 'unique' sentences and 10 million sentence comes from training the model for 10 epochs.

[28] In both cases they are using GloVe vectors and have a TSA.

Authors	Laptop (%)	Restaurant (%)	Fine Tuned	CWR Model	Pre-Trained	TSA
Rietzler et al., 2020	80.23	**87.89**	Yes	BERT_b	Yelp, Amazon	No
Zeng et al., 2019	**82.45**	87.14	Yes	BERT_b	No	Yes
Jiang et al., 2019	-	85.93	Yes	BERT_b	No	Yes
Xu et al., 2019	78.07	84.95	Yes	BERT_b	Yelp , Amazon, SQuAD	No
Zhao et al., 2019	81.35	83.57	No	BERT_b	No	Yes
Song et al., 2019	79.93	83.12	Yes	BERT_b	No	Yes
Huang et al., 2019	80.1	83.0	No	BERT_l	No	Yes
Bert_b=BERT base model, Bert_l=BERT large model (Devlin et al., 2019) TSA=Task Specific Architecture						

Table 5.2: Overview of TDSA methods that use CWR. Results on the Laptop and Restaurant datasets are reporting accuracy scores.

Authors	Laptop (%)	Restaurant (%)
Zhao et al., 2019	75.55	**82.95**
Zeng et al., 2019	**76.02**	82.5

Table 5.3: Top performing non-CWR TDSA methods

Thus to summarise from this prior work, we can determine that:

1. Pre-training is always useful but to be fully utilised it requires fitting on large amounts of pre-training data.
2. Using a TSA with CWR is better than using no TSA with CWR.
3. Using CWR is always better than non-CWR.

The CWR that will be used to investigate the effect it has on the TDSA and text classification methods in this thesis is the ELMo transformer (ET)[29], as it is quicker for both training and inference than the standard LSTM ELMo, and generally better than a Gated CNN version (Peters et al., 2018b). Furthermore, this is used in the *feature based* manner with a weighted combination of it each layers weights. *Feature based* was chosen over *fine-tuning* as *fine-tuning* adds a large number of parameters to the task specific model, which would therefore mean that we are testing not just CWR but also the affect of adding more parameters. The effect of adding more parameters is thus undesirable and not needed, hence a *feature based* approach was chosen. Furthermore, ET was chosen over BERT due to practicalities of training these models at the time of experimentation. Lastly, the point of these experiments is to test the differences between CWR and non-CWR rather than differences in CWR for TDSA. As it has been shown in the prior work that domain specific CWR out perform non-domain specific, the ET model is pre-trained using an LM task for each domain Laptop, Restaurant and Election. For the Laptop dataset the ET model is pre-trained on the Amazon electronics reviews dataset[30] (McAuley et al., 2015), where the ET model is trained on 28,742,985 in domain sentences. The Restaurant dataset uses the 2019 Yelp dataset[31] where the ET model is trained on 27,286,698 in domain sentences. Finally the Election dataset unlike the others trains the ET model from scratch[32] on 9,903,000 in domain tweets[33]. All of the detailed pre-processing, analysis and training of these domain specific ET models can be found here `https://github.com/apmoore1/language-model`. As Rietzler et al. (2020) found that at least 10 million pre-training sentences are required before the CWR models perform any better than their non-domain specific CWR, furthermore the performance of their models saturate after 17 million sentences. This advice from Rietzler et al. (2020) has been followed for the Laptop and Restaurant ET models but not the Election ET

[29]The base ET model can be found here `https://allennlp.org/elmo` named as the 'Transformer ELMo' model.

[30]Can be found here `http://jmcauley.ucsd.edu/data/amazon/`

[31]`https://www.yelp.com/dataset`

[32]Both the Laptop and Restaurant domain specific ET models were first pre-trained on the 1 Billion word corpus (Chelba et al., 2014) before being further trained on their respective domain specific Amazon and Yelp datasets. In comparison the Election domain specific ET model was trained from scratch, as in it was not trained on this 1 Billion word corpus and all parameters were random initialised before training on the MP Twitter data.

[33]These tweets were collected by scraping Tweets that originate from the current MPs of the time.

model. Within the experiment section considering none of the prior work investigated what effect the CWR had on TDSA and text classification methods other than overall Accuracy and macro F1, this will be explored through the recommended metrics and error splits.

5.4.2 Experiments

To fully report the results figure 5.18 shows the different metric scores the CWR TDSA and text classification (*CNN*) models scored on all datasets. The more informative is figure 5.19 which compares the metric scores of the CWR and their respective baseline models. From this figure it can be seen that all of the TDSA models perform better on almost every metric. The *STAC 1* metric on the Election dataset some of the TDSA models perform worse/no better, which may suggest the models are becoming better at the target sentiment relationship modelling and overfitting less to the most frequent sentiment in the sentence as they improve on the *STAC Multi* metric. Also some of the TDSA models perform worse/no better on the *STAC Multi* metric on the Laptop and Restaurant dataset but perform much better on the *STAC 1* metric suggesting the models are overfitting to the most frequent sentiment in the sentence. This initial analysis suggests to some extent that the CWR increase the performance of the TDSA model by exploiting the artefacts of the datasets. This is shown through the fact that the datasets that contain the least number of DS_2 and DS_3 sentences, Laptop and Restaurant, the CWR TDSA models perform a lot better compared to their baseline on the *STAC 1* metric, but not the *STAC Multi*. Whereas the Election dataset that contains more sentences of DS_2 and DS_3 combined than DS_1, the CWR models perform better compared to their baseline on the *STAC Multi* than the *STAC 1* in general.

To investigate this further the heatmaps in figure 5.20 and 5.21[34] show the number of TDSA only CWR models that are statistically significantly better than their baseline, where the former is not correct for multiple significance tests where as the later is using Bonferroni and is aggregated across datasets. It can be seen that for the accuracy metric the CWR improve over the baseline significantly. Macro F1 when corrected for multiple tests it is not shown as significant as seen in figure 5.21. In general and as stated before the CWR models perform significantly better on the *STAC Multi* for the Election dataset but for those that contain far fewer DS_2 and DS_3 samples this is not true. The *STAC 1* metric is similar to the accuracy in that in almost all cases the CWR are significantly better than their baselines. These significant test somewhat agree with the initial analysis that the CWR are exploiting the artefacts of the datasets, as they perform generally no better on the *STAC Multi* metric for datasets that contain far fewer DS_2 and DS_3 samples (Laptop and Restaurant).

To further investigate whether this initial analysis is true the *DS* error split significance heatmaps can be seen in figures 5.22 and 5.23. Also within those heatmaps are the *TSR* error analysis results, which can be used to study if the CWR improve zero shot target and sentiment relation prediction through the *UT* and *USKT* subsets respectively. The results from the *DS* error splits somewhat confirm the initial analysis as the laptop validation results for the DS_2 subset have no significantly better CWR models. Furthermore, out of the nine possible model and dataset combinations only six and five models are

[34]This heatmap does not show the number of baseline models that are significantly better than their CWR model as none were.

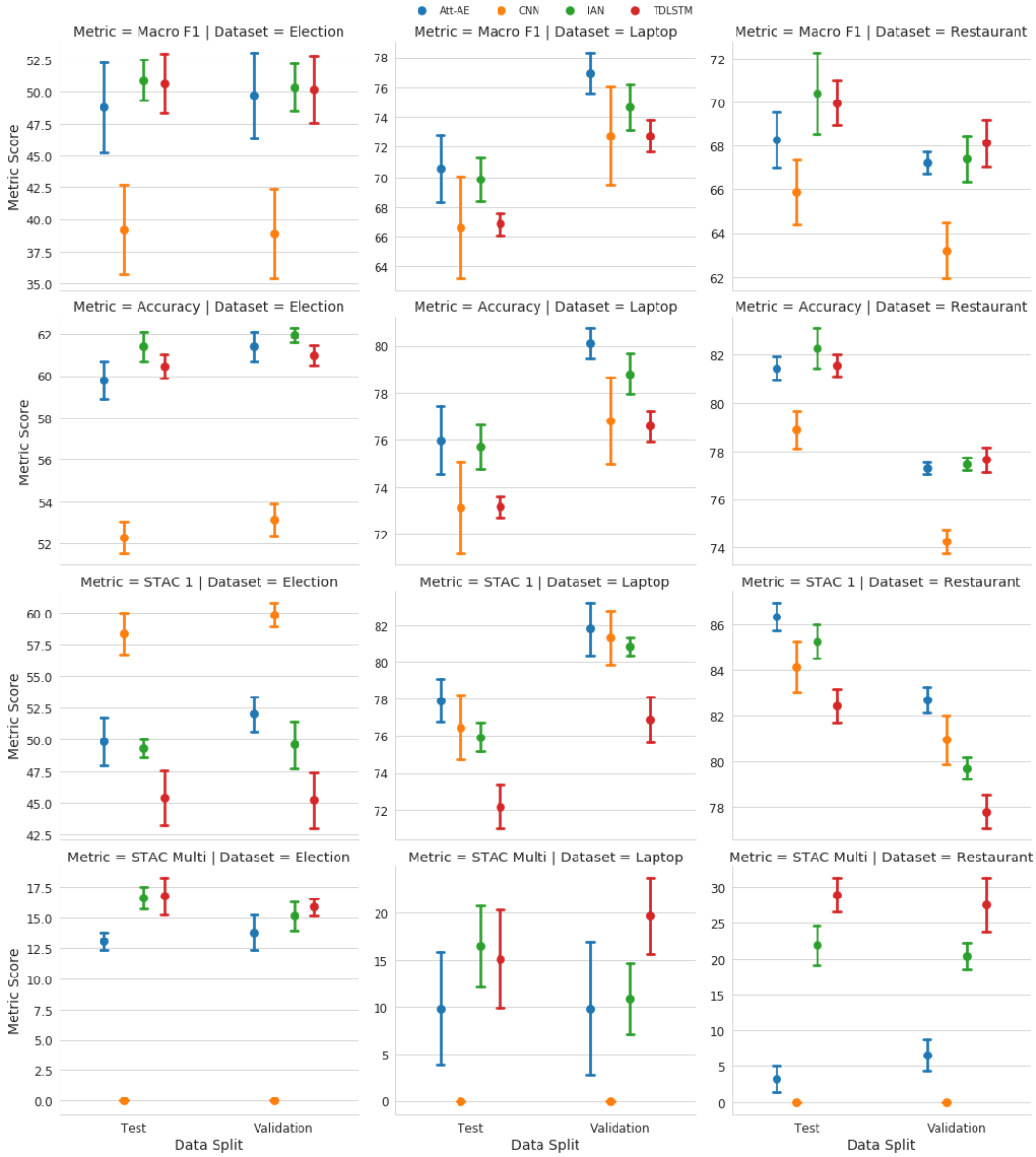


Figure 5.18: Each plot represents the CWR models metric score on the test and validation splits. Columns represent different datasets, rows different metrics.

significantly better than their baseline for the DS_2 subset on the test and validation data splits respectively. This in comparison to almost all CWR models being significantly better on the DS_1 subset.

In comparison to any of the other enhancements, this is the first time that the *UT* and *USKT* subsets have improved significantly. This suggests that contextualising the target representations greatly improves the generalisation of them to new targets that are most likely similar to seen targets.

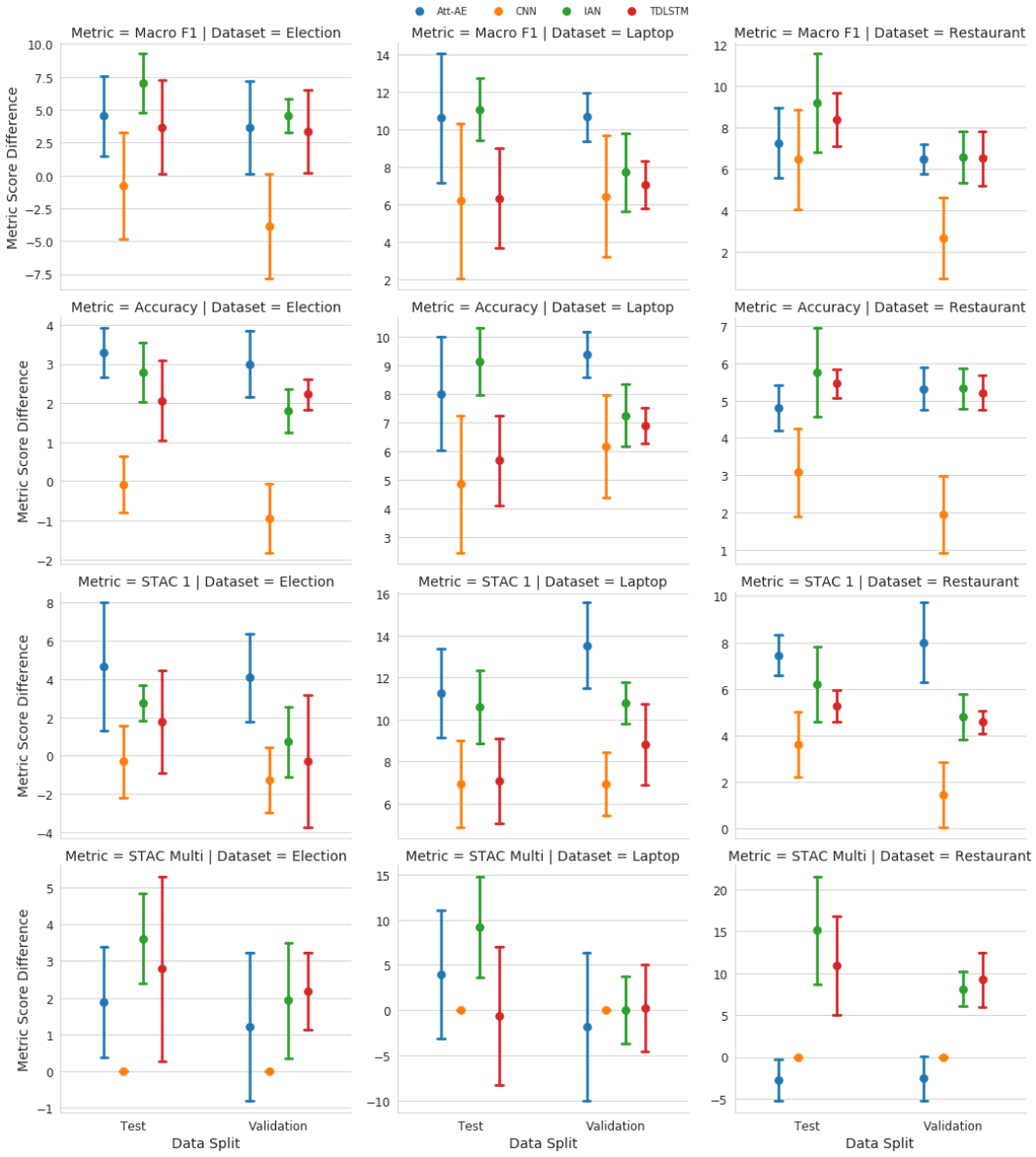


Figure 5.19: Each plot represents the differences between the CWR and the baseline models for the relevant metric score on the test and validation splits. Columns represent different datasets, rows different metrics.

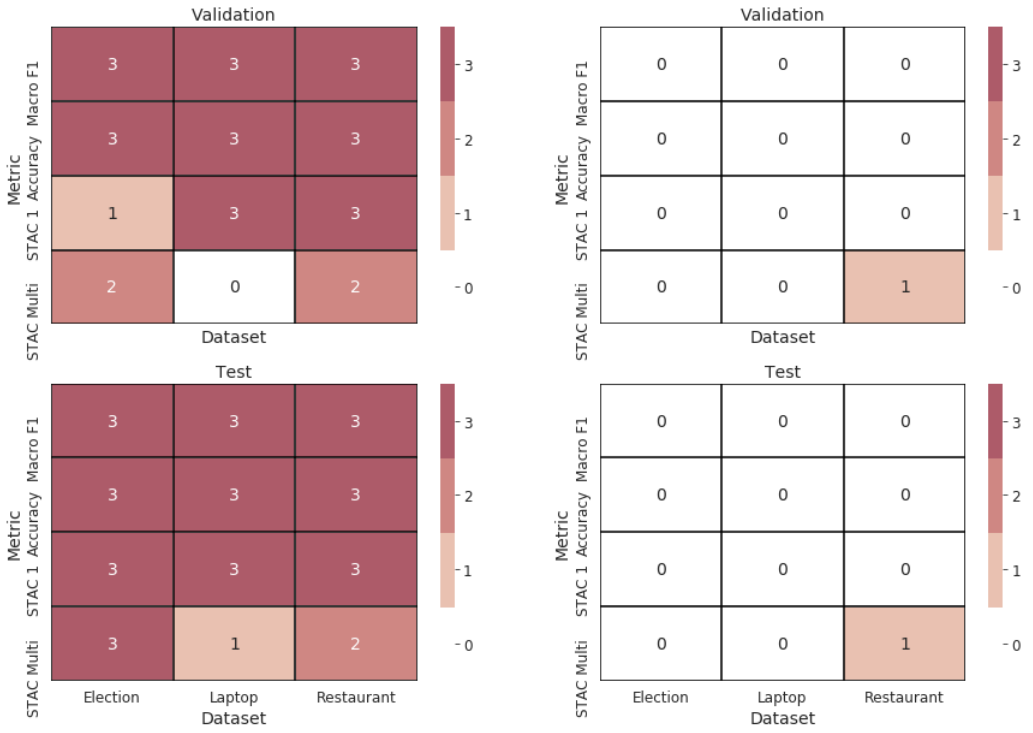


Figure 5.20: The left hand side heatmaps represent the number of CWR models that are statistically significantly better than their baseline equivalents at the 95% confidence level. The right hand side heatmaps represent the number of baseline models that are statistically significantly better than their CWR equivalents at the 95% confidence level.

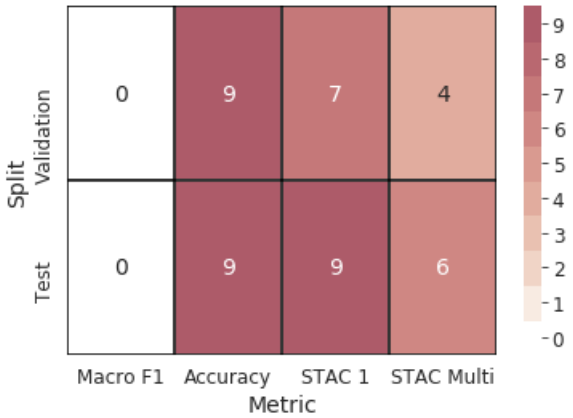


Figure 5.21: The number of CWR models that are statistically significantly better than their baseline equivalents across all datasets at the 95% confidence level. Where the multiple hypothesis tests have been corrected using Bonferroni.

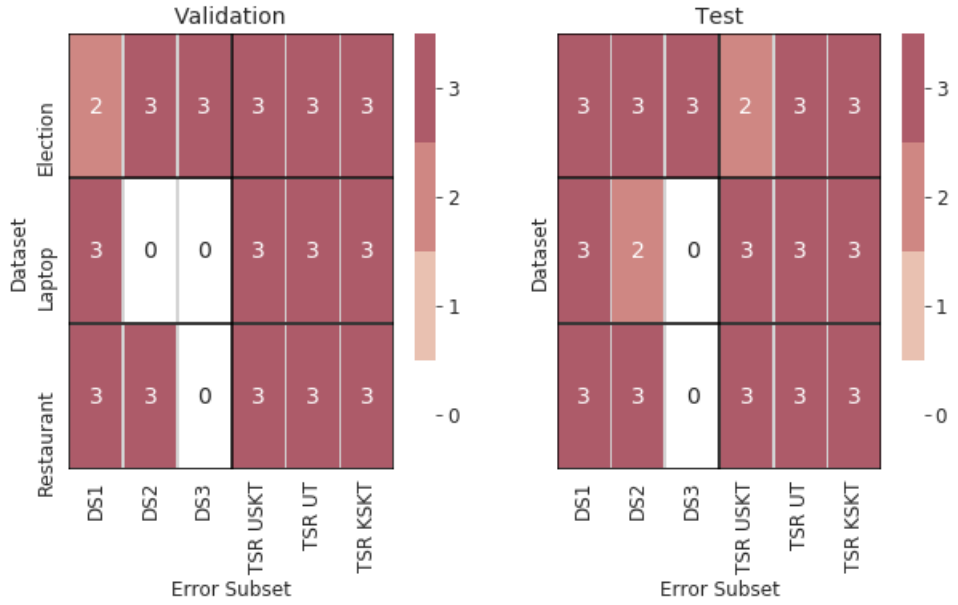


Figure 5.22: Heatmaps represent the number of CWR models that are statistically significantly better than their baseline equivalents at the 95% confidence level based on the accuracy metric.

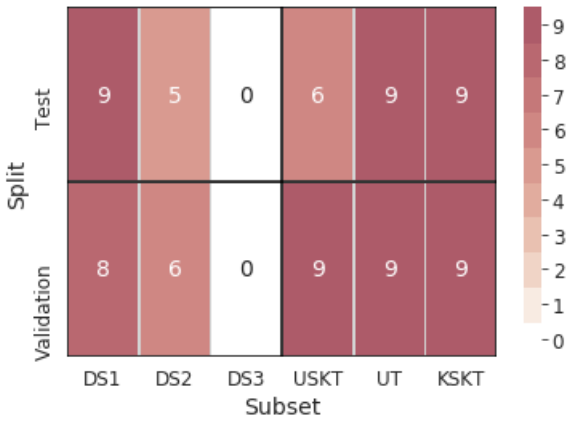


Figure 5.23: The number of CWR models that are statistically significantly better than their baseline equivalents across all datasets at the 95% confidence level based on the accuracy metric. Where the multiple hypothesis tests have been corrected using Bonferroni.

5.4.3 Conclusion

To conclude, in accordance with the existing literature, it is found that through the general accuracy metric, CWR improve TDSA methods significantly compared to using non-CWR (baseline). Unlike the previous work, much greater emphasis has been placed on finding for the first time how CWR improve TDSA models. It has been shown here that the CWR significantly improve the performance of samples that contain unknown targets (UT) or unknown sentiment relationship ($USKT$). Thus suggesting that CWR models will perform better in the real world or low resource setting where a lot of the targets or sentiment relationships are not known. It was also found that the macro F1 scores for the CWR are significantly better than non-CWR, but when correcting for multiple tests was not found to be true. The CWR would appear to only improve significantly for the *STAC Multi* metric and DS_2 subsets when the dataset contains more DS_2 and DS_3 samples like the Election dataset, improving the target sentiment relationship modelling for those TDSA models. Where as the CWR would almost always improve the *STAC 1* metric and accuracy on the DS_1 subset, this suggesting that in some cases the CWR may be overfitting to the most frequent sentiment in the sentence. Lastly, unlike the previous CWR works the overall accuracy results are somewhat lower. This could be due to the fact that a different CWR model was used, ET instead of BERT. Where BERT_b, even though it contains a similar transformer architecture, it has 12 transformer layers compared to ET's 6. Even though the number of layers may not be significant, it has been shown for cross lingual performance in Natural Language Inference to be of great importance (Wang et al., 2020) [35].

5.5 Conclusion

From these three investigations, the position encoding showed successfully that the original hypothesis that adding position information does improve target sentiment relationship modelling, due to the position model's significant improvements on the *STAC Multi* metric and DS_2 subset. Negative results were found for the inter-target encoding experiments as all results showed that the baseline models were no worse if not at times better than their inter-target encoded enhanced models. Furthermore, the original hypothesis that inter-target encoding would improve the model target interaction was not measured using any error split or metric. Lastly, testing on the CWR where the main hypothesis from the previous work was that the models generally improved as shown through either the accuracy or macro F1 score was confirmed in this thesis. Furthermore, the results from the error splits and novel metrics showed for the first time that they significantly improve results for unknown targets (UT) and unknown sentiment relationships ($USKT$). This result was not found in any of the other model enhancements. It was also shown that in general they improve target sentiment relationship modelling through the results of *STAC Multi* and DS_2. However, these results were more convincing on the Election dataset that contained more DS_2 and DS_3 samples, whereas it appeared that the CWR might be overfitting to the overall sentiment in the sentence for datasets that contain less DS_2 and DS_3 samples (Laptop and Restaurant). These three model investigations to a large extent successfully showed the use of this new evaluation methodology and in

[35] See table 4. Depth in table 4 is the same as layers.

the position encoding investigation how they can match the original hypothesis of why position information is important.

In comparison to previous works, this chapter has systematically evaluated multiple TDSA models across multiple datasets, multiple random seeds, and model enhancements evaluating using the appropriate statistical tests. Potentially due to this empirical rigour it has shown negative results with respect to inter-target encoding, which is the opposite of what the original work found (Hazarika et al., 2018), which is believed due to the original work not reporting results across multiple random seeds. For future work it would be of interest to see if increasing the number of DS_2 and DS_3 samples could create models that are better at target sentiment relationship modelling without affecting the DS_1 samples. Additionally, it was shown that the *STAC Multi* metric is still by far the worst performing metric through the CWR experiment, this therefore shows that this metric is what future researchers need to focus on.

Chapter 6

Conclusion

6.1 Thesis Summary

This thesis, in chapter 2, has summarised the literature related to sentiment analysis from a coarse to fine grained perspective, in doing so has created one of, if not, the most extensive reviews of sentiment analysis applied to the English language, that links the different granularities together. In the review a new extended definition of fine grained sentiment analysis has been created, the hextuple, which in comparison to the original (Liu, 2015) removes sentiment ambiguity that was shown through multiple examples and empirical evidence from two existing datasets. Further, parts of the original definition by Liu (2015) that is used within the hextuple, time and sentiment holder, are justified for the first time in this thesis through sentiment ambiguity rather than application uses (Liu, 2015). The literature review concludes with multiple further related topics on implicit sentiment, discourse level considerations, and stance detection, the first two related topics will be revisited in the future work section 6.3 whereas the stance detection topic discussed the similarities between it and sentiment analysis.

The first reproduction study for TDSA was conducted in chapter 3, in which all three chosen papers were successfully reproduced. From these studies it was found for the two NP methods (Vo et al., 2015; Wang et al., 2017a) that scaling features and the C-value for the SVM classifier are significant factors in these methods, and it was noted that Vo et al. (2015) never reported scaling the features and Wang et al. (2017a) did not report the C-value they used. It was further tested that both scaling features and the C-value are not just significant factors for the methods on the one (two) dataset(s) that Vo et al. (2015) (Wang et al. (2017a)) used, but is shown to be significant across six diverse datasets[1]. Thus it was concluded that both scaling and the C-value used should be reported within a structured format like that suggested by Dodge et al. (2019) within Appendix B-D. For the LSTM methods it was shown for at least one metric (macro F1) that they can be statistically significantly affected by random seeds, which is what Reimers et al. (2017) found for neural sequence labelling methods. These findings suggests a possible reason why previous papers that attempted to replicate (Chen et al., 2017) and reproduce (Tay et al., 2018) Tang et al. (2016b) methods could not do so, this problem was the original motivation behind the reproduction of Tang et al. (2016b). From this it was concluded to follow Reimers et al. (2017) advice on reporting the distribution of results from multiple

[1]These six datasets include the datasets from Vo et al. (2015) and Wang et al. (2017a).

runs to allow for better evaluative comparisons and more accurate representation of the method, as single run scores can be misleading.

Additionally, it was found for Vo et al. (2015) methods that larger general word embeddings (GloVe) are at least as good and in Tang et al. (2016b) case were found to be in general better than smaller type and/or task specific embeddings. This suggests that for Vo et al. (2015) methods that once the smaller type and/or task specific embeddings have been trained the methods can be more energy efficient without losing any performance gains. Alternatively it can be seen that smaller type and/or task specific embeddings are not required. These findings are rather restricted findings as they were only tested on the one Twitter dataset (Dong et al., 2014). These findings from the reproduction studies thus answers RQ 1 'What lessons can be learned from reproducing a method within TDSA?'.

Additionally, in chapter 3, the methods reproduced from the chosen studies are then used within the mass evaluation experiments to test the methods' generalisable capabilities. These three diverse sets of methods are then evaluated across six English datasets that differ in type, domain, medium, dataset sizes, and sentiment class distributions. It was found that no one method performs best or could be declared generalisable, but the NP methods performed better across more datasets than the LSTM methods, however the LSTM methods tend to prefer larger datasets. The LSTM methods were shown to be greatly affected by dataset size and/or sentiment class distribution and when in the low resource setting for all datasets the LSTM methods in some cases can only predict the majority class correctly. Thus it was found that the NP methods are the preferred method in low resource or highly unbalanced class distribution settings. When in the low resource setting it was found that sentiment lexicons, for the NP methods, as an inductive bias, can be useful if the lexicon comes from the same type and medium. However when the NP methods were tested in the normal settings the sentiment lexicons were not shown to be useful, this was also found for the dependency parser features in both low and normal settings. By testing the methods across these six datasets it showed that the novel target aware LSTM methods (TDLSTM and TCLSTM) could be beaten by the non-target aware baseline on at least one dataset, and this was never found in the original work (Tang et al., 2016b), as they only evaluated on one dataset. All of these findings from the mass evaluation experiments answer RQ 2 'How generalisable are existing methods within TDSA?'.

In chapter 4 the prior work on error analysis splits were reviewed extensively, from which two new error splits are created: *TSSR* which measures sentiment overfitting to the most frequent sentiment class within the text, and *TSR* which measures generalisation to Unknown Targets (*UT*s) and Unknown Sentiment relationships for Known Targets (*USKT*s). The prior and new error splits are then analysed across three English datasets showcasing the differences between the datasets through the splits, through which it is found that smaller datasets tend to have more *UT*s and *USKT*s. These findings thus show that in the low resource setting a method that can generalise well to *UT* and *USKT* is required to perform well. Lastly, these prior and new splits are summarised describing what they do and what they hypothetically measure. Three different Neural Network (NN) based TDSA methods and one baseline non-target aware NN method are compared across three English datasets and the results are analysed using the splits. From the analysis it is found that the *NT* split does not measure target interaction,

which it was originally hypothesised to measure (Zhang et al., 2019a). Further, it is recommended to use the *TSR* split over a prior works' *n-shot* split due to it measuring the difference between known targets and *UT* better as well as being capable of also measuring *USKT*s. Also the *TSSR* split could not identify sentiment overfitting to the most frequent sentiment class, however a new metric *STAC* was created whereby the metric difference between its two variants *STAC 1* and *STAC Multi* is hypothesised to better measure the sentiment overfitting. The *DS* split was shown to increase in difficulty when more unique sentiments existed within the text, and thus believed to measure target sentiment relationship modelling. However the *DS* split can be influenced by the most frequent sentiment class within the text. Thus the *STAC Multi* metric can be used as a coarse grained version of *DS* whereby this measure is not influenced by the most frequent sentiment class within the text. Therefore this chapter concludes with a set of recommended error splits and metrics that can measure different phenomena within TDSA, forming the new empirical evaluation methodology for TDSA. Thus the chapter answers RQ 3 'What is an appropriate empirical evaluation methodology for TDSA?'

Following the creation of this new empirical evaluation methodology within chapter 4, chapter 5 conducts several case studies to further test the evaluation methodology. These case studies use the same methods and datasets as chapter 4, but each case study enhances the method with a new development. These enhanced methods are then compared to their non-enhanced version using the new evaluation methodology. The findings from these case studies are then compared, where appropriate, to the original hypothesis that were associated with the relevant development. The original hypothesis from the prior works have only ever been justified either through small qualitative case studies, improvements in overall metric scores, such as accuracy on the entire datasets, or through the unsuitable *NT* error split as shown in chapter 4. From the three case studies the position encoding enhancement was shown to match the original hypothesis of improving target sentiment relationship modelling through significant improvements on the *STAC Multi* metric and DS_2 subset. A negative result was found through the inter-target encoding enhancement, whereby the non-enhanced version was no worse and at times better than the enhanced one. Further it would not have been possible to measure the original hypothesis of inter-target encoding as the empirical evaluation methodology cannot measure target interaction. Additionally it is believed that the negative results might be due to the original work by Hazarika et al. (2018) not evaluating their methods rigorously enough, as they only report results from one run, and it has already been shown in chapter 3 that results from different runs can be significantly different. The last case study evaluated the CWR enhancement, where no prior work has shown how they improve over non-CWR apart from through overall metrics like accuracy. The CWR significantly performed better on the *UT* and *USKT* and this was not found for any of the other developments. Additionally, significant improvement also occurred for the *STAC Multi* metric and DS_2 samples suggesting that it improves target sentiment relationship modelling, but these results were more convincing on the Election dataset which contained higher quantities of DS_2 and DS_3 samples. This suggests in part that the CWR enhancement might be overfitting to artefacts within the trained dataset. These case studies have rigorously tested the new empirical evaluation methodology and demonstrated how it can better quantify new developments and how these new developments improve TDSA.

Throughout this thesis each experiment has been conducted within a rigorous experimental setup, which includes significance testing and using a correction procedure, Bonferroni, when appropriate.

6.2 Research Limitations

Given that the thesis summary has stated how the three research questions have been addressed within this thesis, this section will state the limitations of the answers to these research questions.

- **What lessons can be learned from reproducing a method within TDSA?** (RQ 1)
  The answer to this has only tested for two parameters for the NP methods, scaling features and C-values within the SVM, and one parameter, random seeds, within the neural LSTM methods. There are many more parameters that could have been tested, for instance not having a standard train, validation, test setup for the neural methods and only having a training and test setup. It has already been shown that there can be differences between methods when using different train and test splits as shown for POS tagging (Gorman et al., 2019) and NER (Moss et al., 2019). Thus it is likely that by not having a standard validation split within the already standard train test splits will create reproducibility problems. However the parameters that were explored were justified as it was already shown by Reimers et al. (2017) that it was important to report multiple runs of a neural based method. The C-values were tuned for one of the NP methods (Vo et al., 2015) thus it seemed logical to investigate if this was important for the other NP method (Wang et al., 2017a). Also scaling features was used within the codebase of one of the NP methods (Wang et al., 2017a), but it was not stated to be used for the other NP method (Vo et al., 2015). Thus there are likely many parameters that are known and unknown that would affect reproducibility, but in this thesis only three key parameters were explored.

- **How generalisable are existing methods within TDSA?** (RQ 2)
  There are two large limitations with this work. The first being the set of methods that were evaluated, which were in general Neural Pooling (NP) and LSTM methods. Even though they represent two different large sets of methods non-neural and neural, however there are a number of different non-neural and neural methods e.g. transformers (Vaswani et al., 2017) which have become very popular recently and unlike LSTM does not have a strong dependency/bias between its past encoded tokens. Further, both the NP and LSTM methods use the same type of input representation, the word embedding, where for the non-neural methods a bag of words input would be appropriate and could have created another strong and interesting baseline to compare too. The second limitation is that all datasets are in the same language, English, thus the findings are currently restricted to the English language only.

- **What is an appropriate empirical evaluation methodology for TDSA?** (RQ 3)

The empirical evaluation methodology is limited to testing phenomena that can be automatically created but has not yet been tested for linguistic phenomena. There are many linguistic phenomena that would be of interest to explore such as those suggested by Barnes et al. (2019) that includes negation, amplifier, emoji, and many more.

6.3 Future Work

In this section, the future work will be stated that is based upon the findings within this thesis and the literature review. However, it is worth noting that there has been a very comprehensive survey on the future of sentiment analysis by Poria et al. (2020) that explores many worthwhile avenues for sentiment analysis in general.

- As found within chapter 3, the generalisation experiments showed the neural methods under perform on small and un-balanced datasets. Thus testing different methods such as transfer learning from language models, which has been demonstrated to be successful in low resource settings (Howard et al., 2018), and transfer learning from document sentiment analysis which has already been shown useful on un-balanced datasets (He et al., 2018b) would be of interest.

- Through the error analysis studies in chapters 4 and 5 it was found that many methods did not perform well on target sentiment relationship modelling, measured through the *DS* split and *STAC Multi* metric. Thus finding a way to improve this would be of use.

- Also through the error analysis, within chapter 4 and 5, Unknown Targets (*UT*s) and especially Unknown Sentiment Known Targets (*USKT*s) performed poorly in comparison to Known Sentiment Known Targets (KSKT). Again finding a way to improve on samples within the *UT* and *USKT* would be a potentially fruitful future direction, more so as these subsets of data tend to occur more within low resource settings as shown by chapter 4. A proposed approach to this problem would be to better model the target itself as done within He et al. (2018a).

- The literature review highlighted, within section 2.5.2, the importance of discourse information within fine grained sentiment analysis. It would be of interest to study the importance of discourse information within TDSA. One area of discourse information that could be of use is co-reference resolution. This is motivated by the fact that Kessler et al. (2009) found that 14% of targets are pronouns[2], thus if recent TDSA methods perform differently on these pronoun targets it could motivate the need for incorporating co-reference information into the TDSA method. Further, alternative discourse information could be of use too so that methods take into account more than the current sentence, as Kessler et al. (2010) found that 9% of sentiment expressions are not within the same sentence as the target they affect.

- Poria et al. (2020) indicates that commonsense information could help improve implicit and factual sentiment analysis and we agree with Poria et al. (2020) that this would be a fruitful avenue of future research. In the pursuit of this research

[2]The corpus this was performed on was an early version of JDPA corpus (Kessler et al., 2010).

the review on implicit and factual sentiment analysis, within section 2.5.1, should be of use.

Appendix A

Reproducibility and Generalisability of TDSA Methods

A.1 Tables

		Fold				
Method	Embedding	1	2	3	4	5
TI	w2v	**0.0187**	**0.0492**	0.1708	0.1891	0.4314
	SSWE	0.1132	0.0533	0.0567	**0.0099**	0.4532
	GloVe	0.7992	0.9976	0.6734	0.4927	0.8801
TDM	w2v	0.4774	**0.0046**	**0.0023**	**0.0328**	**0.0124**
	SSWE	0.0858	0.0717	0.2807	0.8586	0.1042
	GloVe	0.8991	0.3767	0.1138	0.8357	0.2367
TD	w2v	0.1236	**0.0158**	**0.0274**	0.0533	0.6894
	SSWE	**0.0103**	**0.0117**	0.0652	0.1148	0.1054
	GloVe	0.7350	0.8223	0.1859	0.6779	0.9725
TDP	w2v	0.1314	0.1086	0.1825	0.2366	0.5573
	SSWE	0.0798	**0.0027**	**0.0311**	0.2026	0.4478
	GloVe	0.5684	0.4610	0.0743	0.6331	0.8284

Table A.1: P-values for the accuracy metric, testing if SSWE + w2v embedding is significantly better than the other embeddings for each method on each fold of the five-fold cross validation. The five folds came from the Dong et al. (2014) Twitter training dataset. All p-values that are significant ≤ 0.05 are in **bold**.

		Fold				
Method	Embedding	1	2	3	4	5
TI	w2v	**0.0028**	**0.0057**	**0.0362**	**0.0192**	0.1254
	SSWE	0.1101	**0.0023**	**0.0348**	**0.0012**	0.1321
	GloVe	0.9057	0.9990	0.8106	0.6867	0.8499
TDM	w2v	0.2474	**0.0024**	**0.0000**	**0.0044**	**0.0018**
	SSWE	**0.0429**	**0.0241**	0.1018	0.5029	0.0754
	GloVe	0.8688	0.4317	0.0740	0.8162	0.2744
TD	w2v	**0.0108**	**0.0063**	**0.0031**	**0.0052**	0.3455
	SSWE	**0.0012**	**0.0008**	**0.0280**	**0.0188**	0.0738
	GloVe	0.6811	0.8059	0.2267	0.7103	0.9657
TDP	w2v	**0.0194**	**0.0382**	0.0650	0.0961	0.3623
	SSWE	**0.0221**	**0.0003**	**0.0151**	0.0819	0.3732
	GloVe	0.5240	0.4813	0.0969	0.7047	0.8225

Table A.2: P-values for the macro F1 metric, testing if SSWE + w2v embedding is significantly better than the other embeddings for each method on each fold of the five-fold cross validation. The five folds came from the Dong et al. (2014) Twitter training dataset. All p-values that are significant ≤ 0.05 are in **bold**.

		Fold				
Method	Embedding	1	2	3	4	5
TI	w2v	**0.0112**	**0.0002**	0.1359	0.2858	0.0954
	SSWE	**0.0403**	**0.0001**	**0.0485**	**0.0375**	0.1191
	SSWE + w2v	0.2157	**0.0034**	0.3456	0.5330	0.1276
TDM	w2v	0.1051	**0.0469**	0.1746	**0.0125**	0.2010
	SSWE	**0.0176**	0.2131	0.7693	0.4632	0.4148
	SSWE + w2v	0.1169	0.6399	0.8943	0.1793	0.7803
TD	w2v	0.0568	**0.0034**	0.2774	**0.0486**	0.0679
	SSWE	**0.0127**	**0.0072**	0.3882	0.0939	**0.0041**
	SSWE + w2v	0.2800	0.1882	0.8277	0.3421	**0.0351**
TDP	w2v	0.1747	0.2198	0.7834	0.1974	0.1889
	SSWE	0.1218	**0.0275**	0.4635	0.1772	0.1763
	SSWE + w2v	0.4610	0.5737	0.9357	0.3973	0.1862

Table A.3: P-values for the accuracy metric, testing if GloVe embedding is significantly better than the other embeddings for each method on each fold of the five-fold cross validation. The five folds came from the Dong et al. (2014) Twitter training dataset. All p-values that are significant ≤ 0.05 are in **bold**.

		Fold				
Method	Embedding	1	2	3	4	5
TI	w2v	**0.0001**	**0.0000**	**0.0125**	**0.0246**	**0.0278**
	SSWE	**0.0120**	**0.0000**	**0.0107**	**0.0014**	**0.0350**
	SSWE + w2v	0.0904	**0.0014**	0.1828	0.3076	0.1542
TDM	w2v	0.0523	**0.0159**	0.0520	**0.0033**	0.0635
	SSWE	**0.0099**	0.0889	0.6647	0.2007	0.3209
	SSWE + w2v	0.1325	0.5576	0.9246	0.1876	0.7198
TD	w2v	**0.0078**	**0.0021**	0.0640	**0.0065**	**0.0164**
	SSWE	**0.0057**	**0.0015**	0.2117	**0.0226**	**0.0017**
	SSWE + w2v	0.3191	0.1955	0.7637	0.2803	**0.0334**
TDP	w2v	0.0575	0.1054	0.5351	0.0542	0.1034
	SSWE	0.0582	**0.0064**	0.2749	**0.0461**	0.1327
	SSWE + w2v	0.4589	0.5115	0.9006	0.2989	0.1690

Table A.4: P-values for the macro F1 metric, testing if GloVe embedding is significantly better than the other embeddings for each method on each fold of the five-fold cross validation. The five folds came from the Dong et al. (2014) Twitter training dataset. All p-values that are significant ≤ 0.05 are in **bold**.

Embedding	Metric	LSTM	TDLSTM	TCLSTM
SSWE	Accuracy	60.13 (1.24)	66.38 (1.18)	65.30 (1.09)
	F1	56.13 (2.25)	62.63 (1.53)	61.14 (1.51)
Twitter 50	Accuracy	59.13 (0.92)	64.36 (1.33)	63.94 (0.77)
	F1	54.62 (2.47)	60.38 (1.74)	59.05 (1.56)
Twitter 100	Accuracy	60.76 (0.91)	65.29 (0.96)	65.62 (1.01)
	F1	56.23 (2.31)	61.54 (1.50)	61.20 (1.52)
Twitter 200	Accuracy	61.08 (1.41)	66.44 (0.92)	66.35 (0.94)
	F1	56.14 (2.73)	63.40 (1.51)	62.32 (1.38)
GloVe 300	Accuracy	**63.65** (0.94)	**68.29** (0.66)	**67.86** (1.29)
	F1	**60.32** (1.48)	**65.16** (0.87)	**64.35** (1.60)

Table A.5: Validation set mean (standard deviation) results on the Dong et al. (2014) Twitter dataset, across various embeddings and methods. The **bold** values indicate the best embedding score for each method and metric.

Method	Embedding	Accuracy	F1
LSTM	SSWE	0.0745	0.0509
	Twitter 50	**0.0392**	**0.0211**
	Twitter 100	0.1400	0.1442
	Twitter 200	0.1032	**0.0392**
TDLSTM	SSWE	**0.0079**	**0.0010**
	Twitter 50	**0.0002**	**0.0000**
	Twitter 100	**0.0039**	**0.0006**
	Twitter 200	**0.0097**	**0.0083**
TCLSTM	SSWE	**0.0057**	**0.0013**
	Twitter 50	**0.0001**	**0.0004**
	Twitter 100	**0.0176**	**0.0100**
	Twitter 200	0.0958	0.0509

Table A.6: P-values for the one sided significant test on the test set comparing the GloVe embeddings to the other embeddings for each metric and method. The significant test compared the median best run from the 20 runs that each method, embedding, and metric produced. The **bold** values indicate all p-values that are significant (≤ 0.05).

Method	Embedding	Accuracy	F1
LSTM	SSWE	0.8901	0.4171
	Twitter 50	0.4878	0.9470
	Twitter 100	0.5957	0.9509
	Twitter 200	0.7947	0.9560
TDLSTM	SSWE	0.9885	0.8986
	Twitter 50	0.2894	0.9415
	Twitter 100	0.9149	0.4421
	Twitter 200	0.9023	0.6515
TCLSTM	SSWE	0.9765	0.9444
	Twitter 50	0.3615	0.3511
	Twitter 100	0.2046	0.9283
	Twitter 200	0.3832	0.3809

Table A.7: P-values for the one sided significant test on the validation set comparing the GloVe embeddings to the other embeddings for each metric and method. The significant test compared the median best run from the 20 runs that each method, embedding, and metric produced.

A.2 Figures

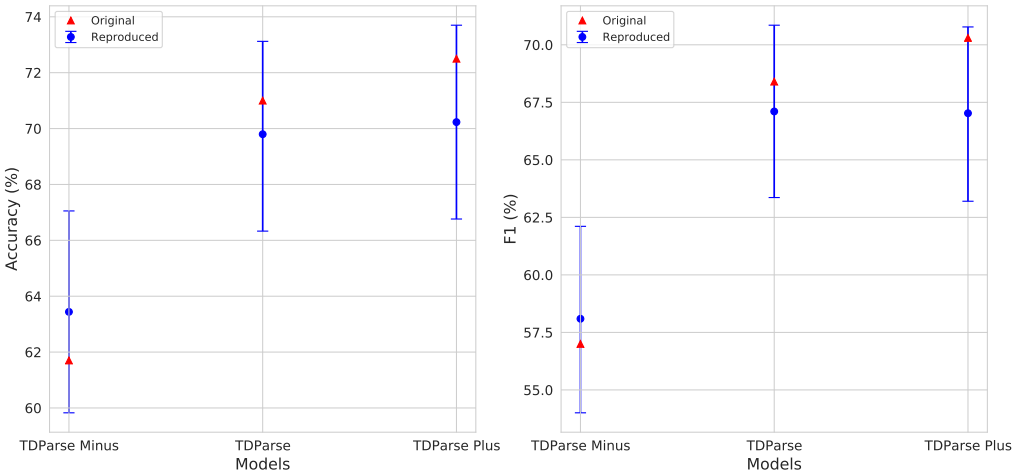


Figure A.1: Using the C-values optimised for macro F1 metric, the confidence intervals for the two tailed test on the Dong et al. (2014) test set.

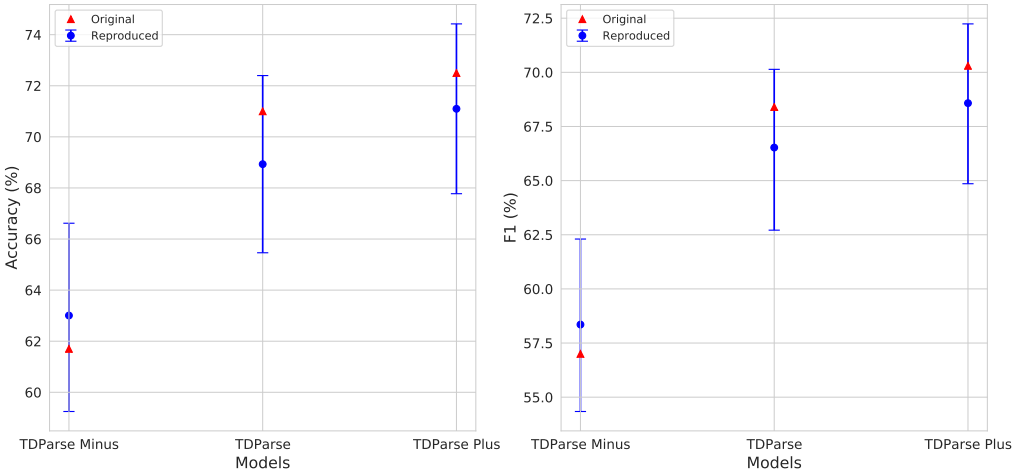


Figure A.2: Using the C-values optimised for macro F1 metric with the original MinMax scaling range of Wang et al. (2017a), the confidence intervals for the two tailed test on the Dong et al. (2014) test set.

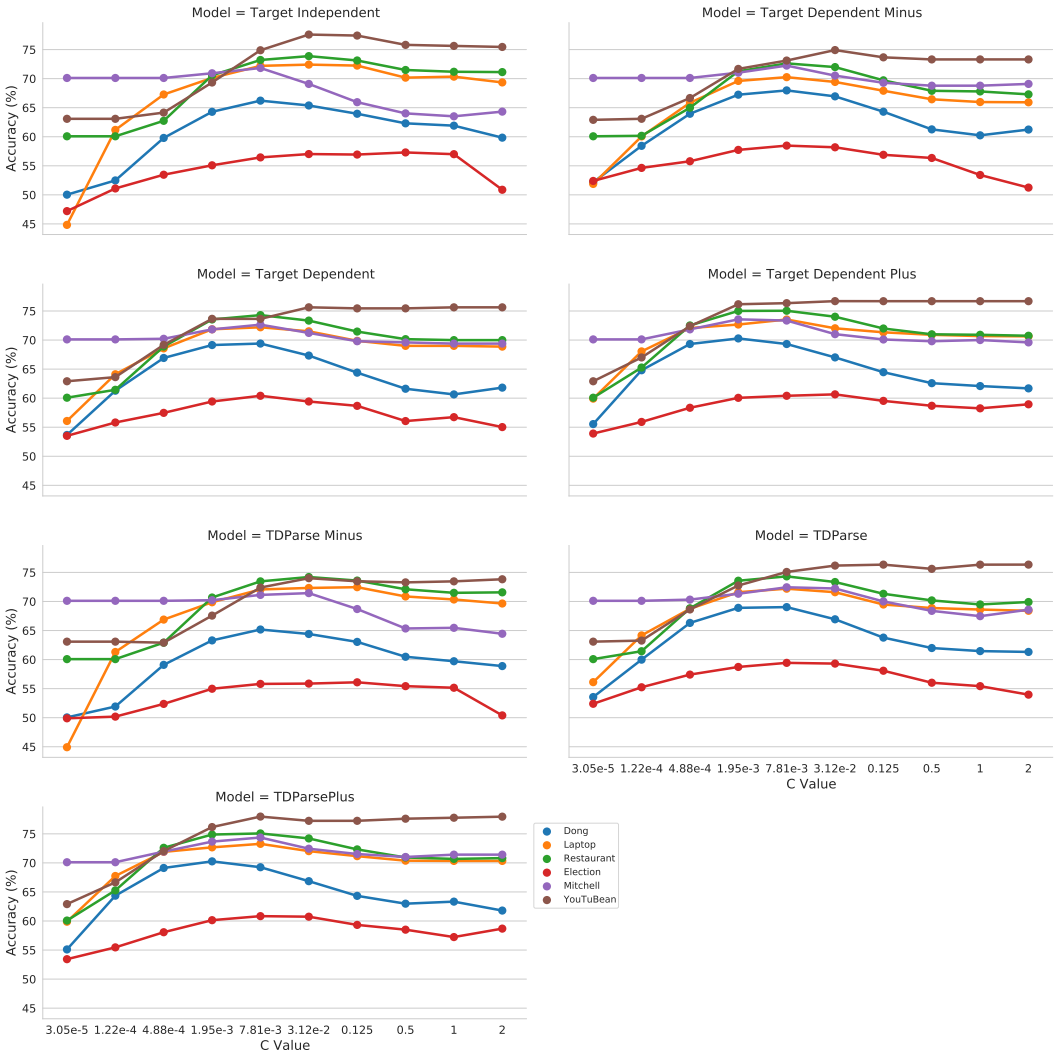


Figure A.3: The mean accuracy from five-fold cross validation on the training set for each C-value.

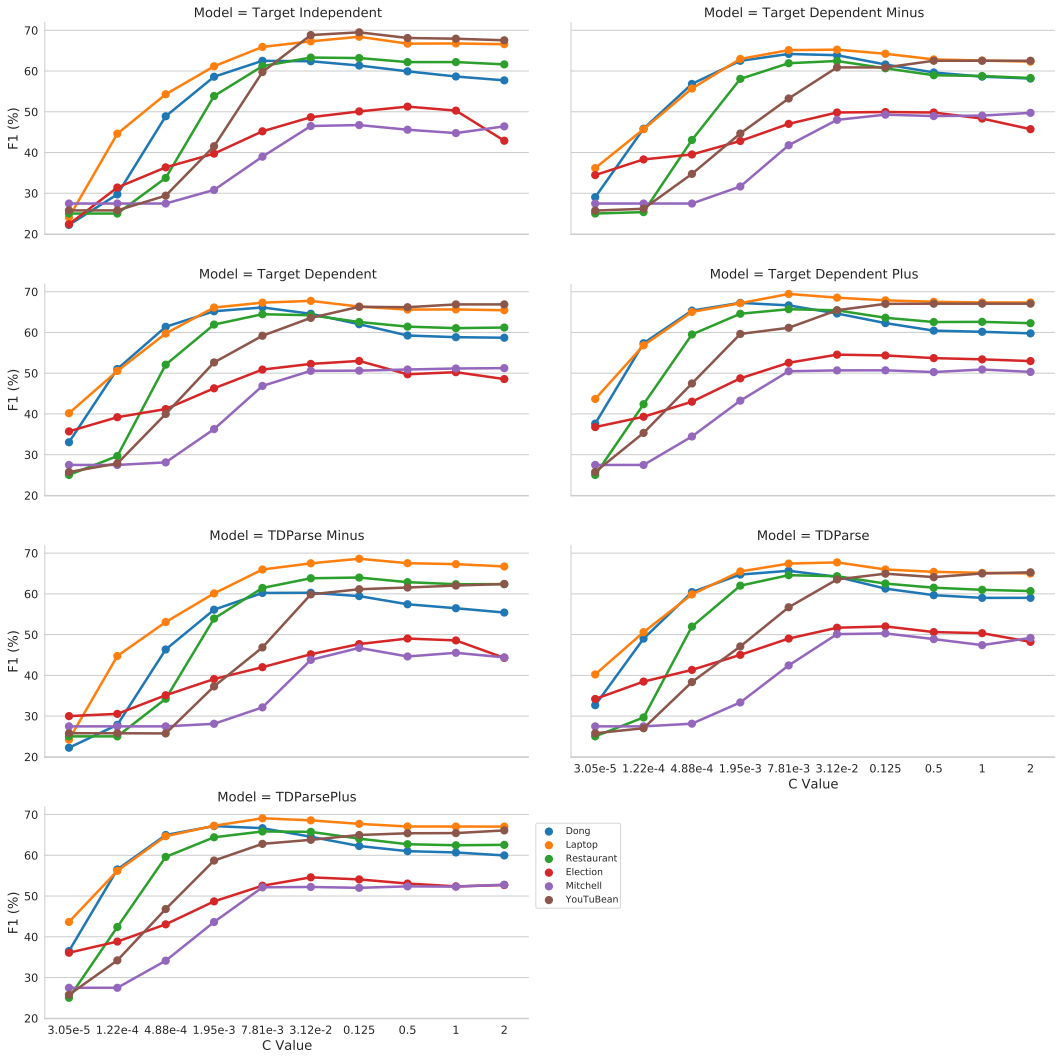


Figure A.4: The mean macro F1 from five-fold cross validation on the training set for each C-value.

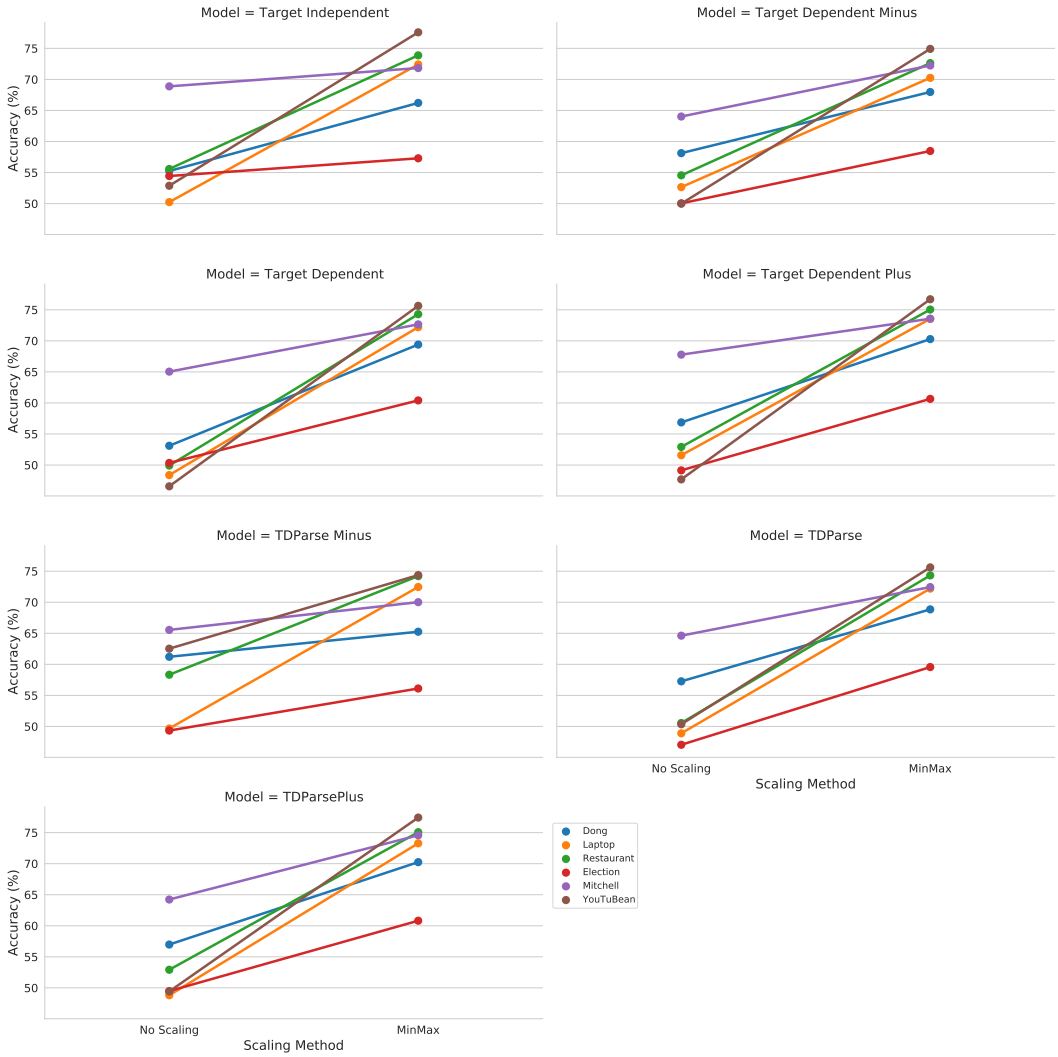


Figure A.5: The mean accuracy from five-fold cross validation on the training set for the two scaling methods.

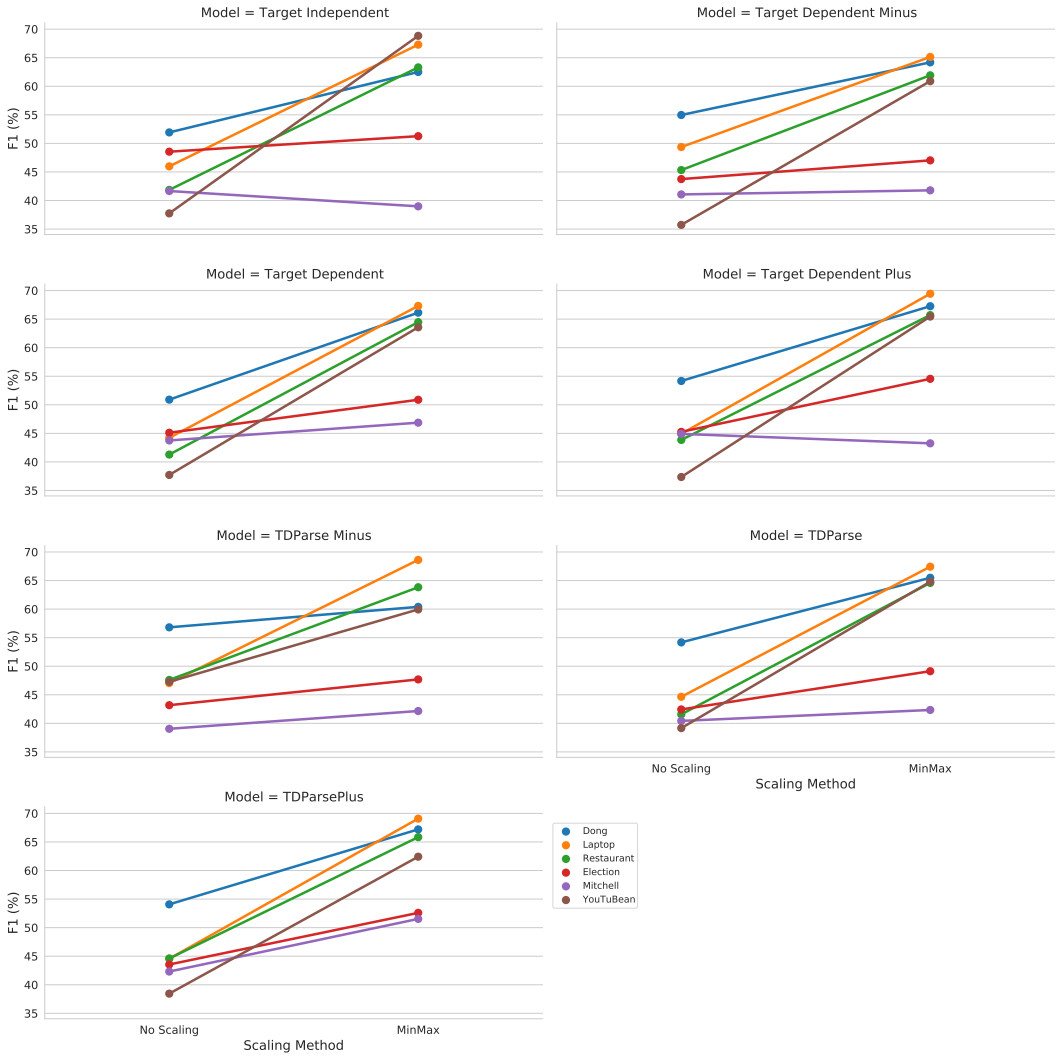


Figure A.6: The mean macro F1 from five-fold cross validation on the training set for the two scaling methods.

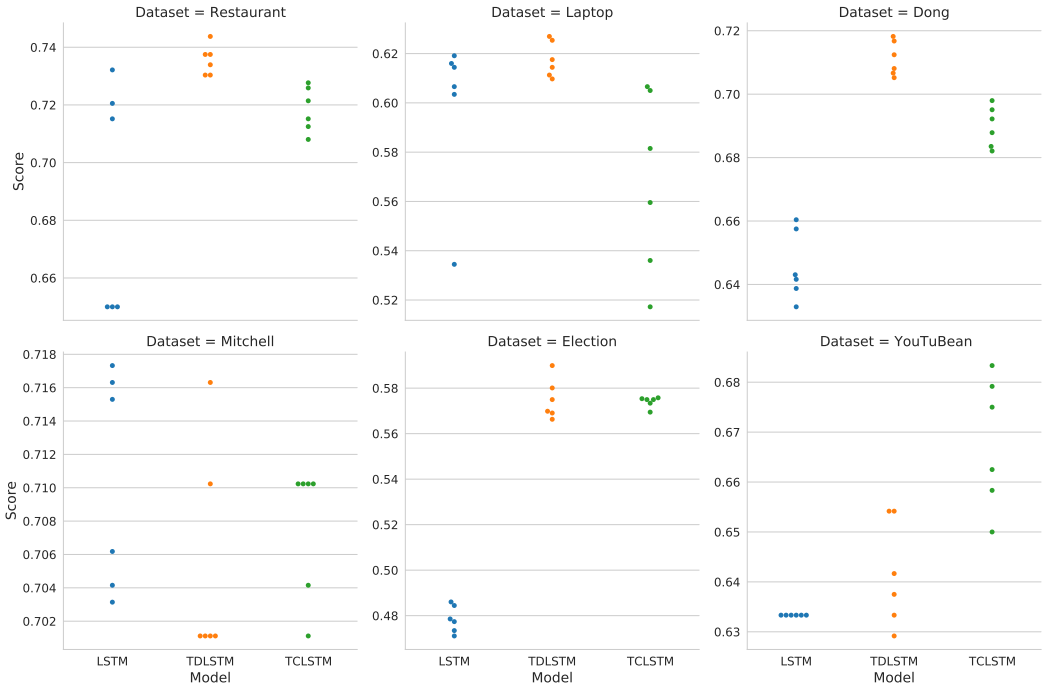


Figure A.7: Distribution of accuracy scores from the six runs for each LSTM method and test dataset.

Appendix B

Improving Experimental Methodology for TDSA

B.1 Tables

Hyperparameter	Value
embedding	GloVe
embedding trainable	False
number of epcohs	100
patience	10
metric early stopping monitored	Accuracy
batch size	32
dropout	0.5
learning rate optimiser	Adam
learning rate	0.001
regularisation type	L2
regularisation value	0.0001

Table B.1: Default hyperparameters.

	Dataset	Election			Laptop			Restaurant		
	Split	OT	ST	V	OT	ST	V	OT	ST	V
Data Split	Data Subset									
TRS	*KTKS*	82.4	80.8	80.9	47.3	42.5	58.6	63.2	60.4	67.9
	USKT	3.7	4.1	4.4	10.5	10.8	8.4	5.4	6.2	5.8
	UT	13.9	15.1	14.7	42.2	46.7	33.0	31.4	33.4	26.3
n-shot	*zero-shot*	13.9	15.1	14.7	42.2	46.7	33.0	31.4	33.4	26.3
	low-shot	29.1	28.3	28.9	19.6	19.9	23.6	23.8	23.1	25.9
	med-shot	29.0	28.7	28.8	19.3	17.7	23.5	23.0	22.4	29.5
	high-shot	28.0	28.0	27.6	19.0	15.7	19.9	21.7	21.1	18.3
OT = original training with test dataset, ST = split training with test dataset V = split training with validation dataset										

Table B.2: Differences between different dataset combinations with respect to global error splits.

	Dataset	Election		Laptop		Restaurant	
	Split	T	V	T	V	T	V
Data Split	Data Subset						
DS_i	DS_1	45.8	44.6	83.9	80.4	79.6	72.0
	DS_2	46.5	47.2	14.7	19.2	20.1	25.3
	DS_3	7.7	8.2	1.4	0.5	0.3	2.7
NT	*1-target*	4.0	4.3	40.6	38.0	25.4	26.4
	low-targets	47.9	46.8	32.0	33.7	34.3	31.5
	med-targets	28.7	25.4	15.5	17.0	30.6	22.7
	high-targets	19.5	23.5	11.9	11.2	9.6	19.4
TSSR	*1-TSSR*	4.0	4.3	40.6	38.0	25.4	26.4
	1-multi-TSSR	41.8	40.2	43.3	42.3	54.2	45.6
	high-TSSR	25.8	25.7	5.0	7.8	8.8	12.5
	low-TSSR	28.4	29.7	11.1	11.8	11.6	15.5
T = test dataset, V = validation dataset							

Table B.3: Differences between the test and validation datasets with respect to local error splits.

		Model			
Metric	Dataset	Att-AE	CNN	IAN	TDLSTM
Accuracy	Election	58.42 (0.40)	54.07 (0.56)	**60.14 (0.40)**	58.73 (0.38)
	Laptop	70.74 (0.75)	70.65 (0.68)	**71.57 (0.64)**	69.69 (0.63)
	Restaurant	71.99 (0.45)	72.31 (0.69)	72.13 (0.57)	**72.43 (0.46)**
Macro F1	Election	46.05 (1.85)	42.74 (2.09)	45.80 (1.48)	**46.83 (1.80)**
	Laptop	66.28 (1.21)	66.32 (0.96)	**66.94 (1.32)**	65.74 (1.00)
	Restaurant	60.77 (0.67)	60.51 (1.20)	60.84 (1.08)	**61.63 (0.87)**

Table B.4: Mean and standard deviation from running each model 8 times on the datasets validation split.

		Model			
Metric	Dataset	Att-AE	CNN	IAN	TDLSTM
Accuracy	Election	56.49 (0.76)	52.35 (0.69)	**58.60 (0.36)**	58.39 (0.74)
	Laptop	67.99 (1.30)	**68.26 (0.69)**	66.58 (0.53)	67.46 (1.60)
	Restaurant	**76.62 (0.59)**	75.81 (0.55)	76.51 (1.49)	76.09 (0.62)
Macro F1	Election	44.23 (1.81)	39.98 (2.20)	43.90 (1.52)	**46.95 (2.33)**
	Laptop	59.97 (2.29)	60.43 (1.36)	58.78 (0.62)	**60.52 (2.63)**
	Restaurant	61.01 (1.47)	59.40 (1.52)	61.22 (2.85)	**61.59 (1.57)**

Table B.5: Mean and standard deviation from running each model 8 times on the datasets test split.

Name	Split	No. Sents(t)	No. Targs (Uniq)	ATS(t)	POS (%)	NEU (%)	NEG (%)
Laptop	Train	1051	1661 (739)	1.58	695 (41.84)	319 (19.21)	647 (38.95)
	Val	411	652 (368)	1.59	292 (44.79)	141 (21.63)	219 (33.59)
	Test	411	638 (389)	1.55	341 (53.45)	169 (26.49)	128 (20.06)
Election	Train	2319	6811 (1496)	2.94	1014 (14.89)	2645 (38.83)	3152 (46.28)
	Val	863	2547 (741)	2.95	352 (13.82)	970 (38.08)	1225 (48.1)
	Test	863	2541 (751)	2.94	378 (14.88)	957 (37.66)	1206 (47.46)
Restaurant	Train	1378	2490 (914)	1.81	1489 (59.8)	422 (16.95)	579 (23.25)
	Val	600	1112 (480)	1.85	675 (60.7)	211 (18.97)	226 (20.32)
	Test	600	1120 (520)	1.87	728 (65.0)	196 (17.5)	196 (17.5)

No. Sents(t)=number of sentences that contain a target, No. Targs (Uniq)=Number of (unique) targets (all targets are lower cased), ATS(t)=Average Target per Sentence where the sentences must contain a target, LABEL (%)=Number of LABEL samples (percentage of LABEL samples).

Table B.6: Dataset statistics for each split for all datasets.

B.2 Figures

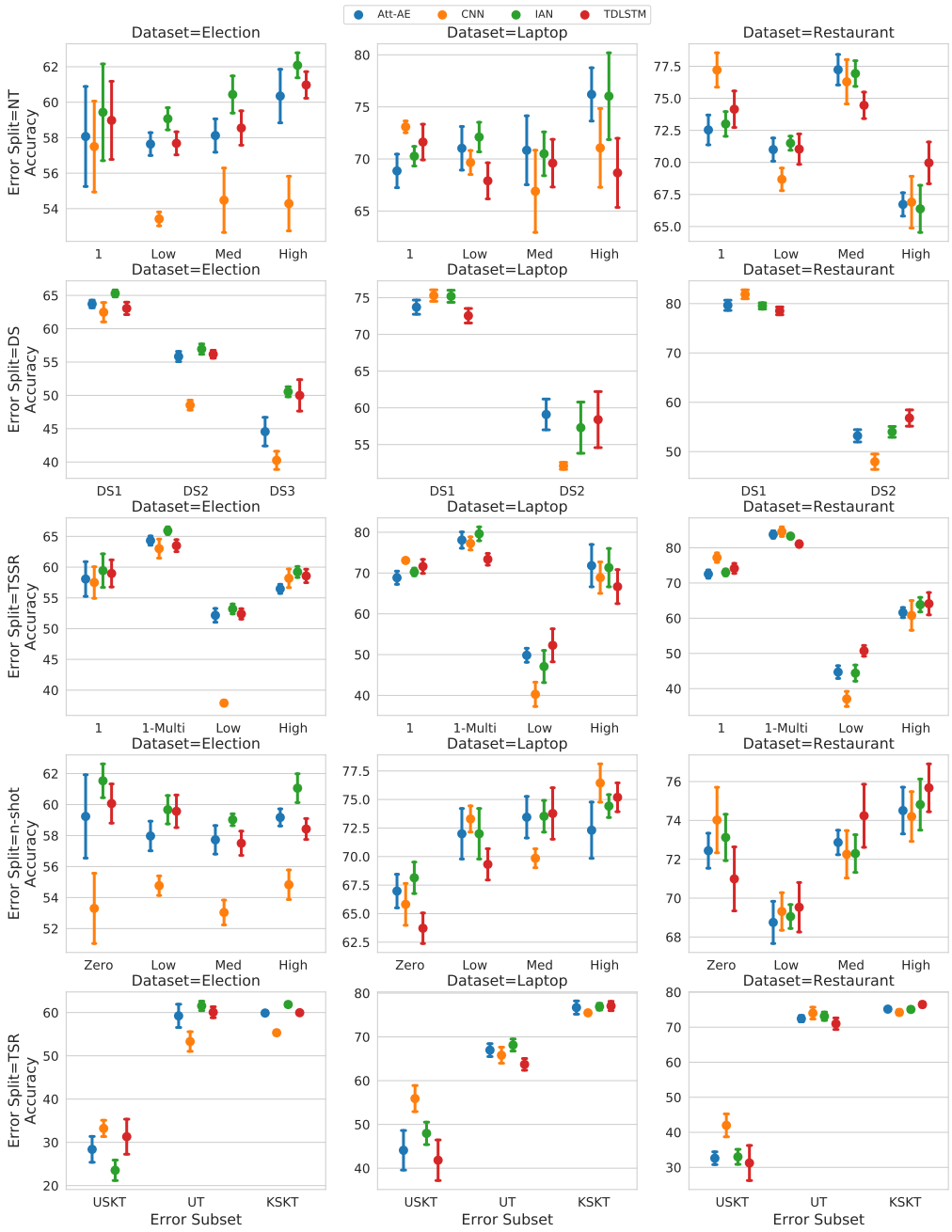


Figure B.1: The mean and standard deviation error bars for each error subset within all of the error splits on the validation split across all datasets.

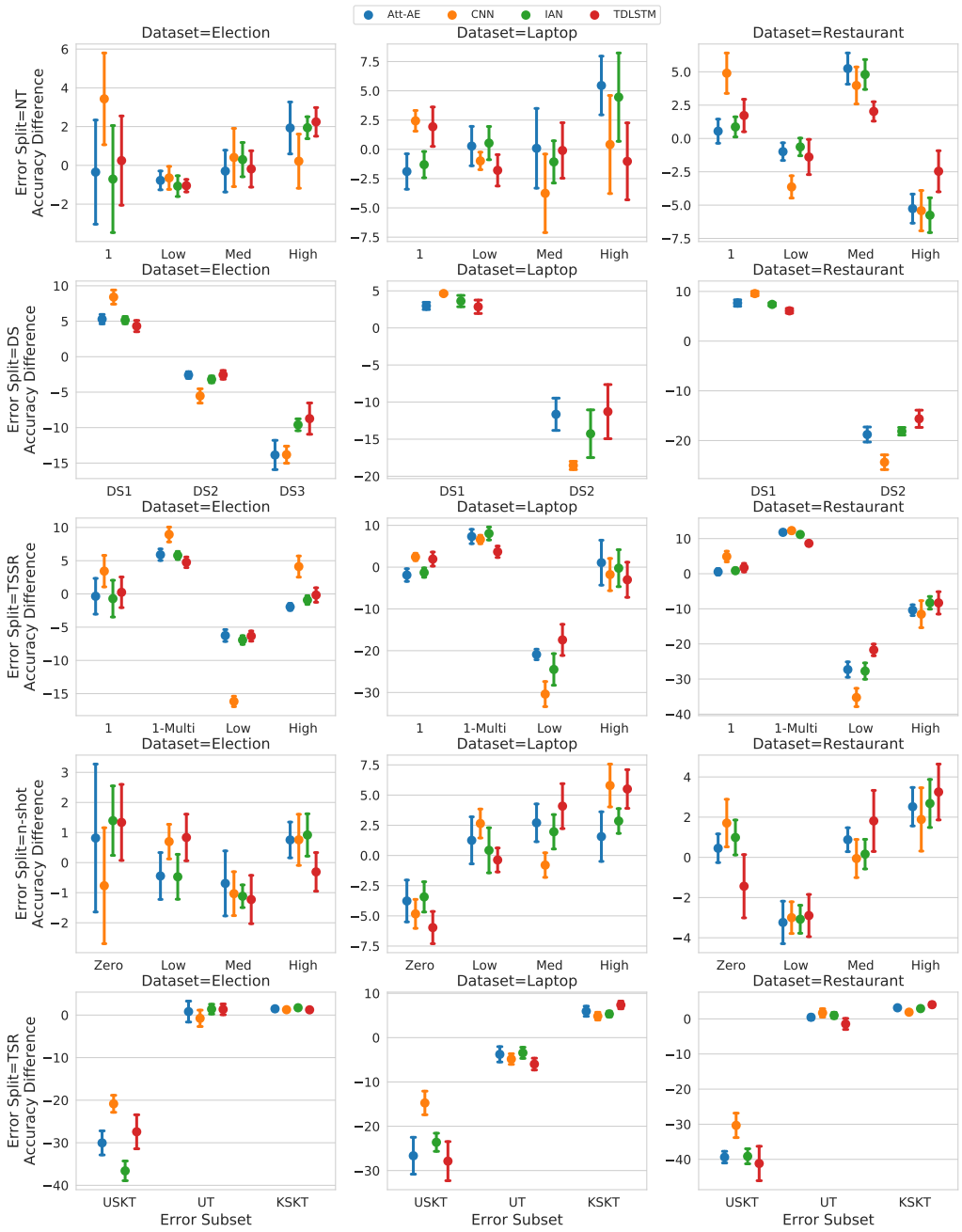


Figure B.2: The mean and standard deviation error bars for the difference between the overall accuracy and the accuracy from each error subset within all of the error splits on the validation split across all datasets.

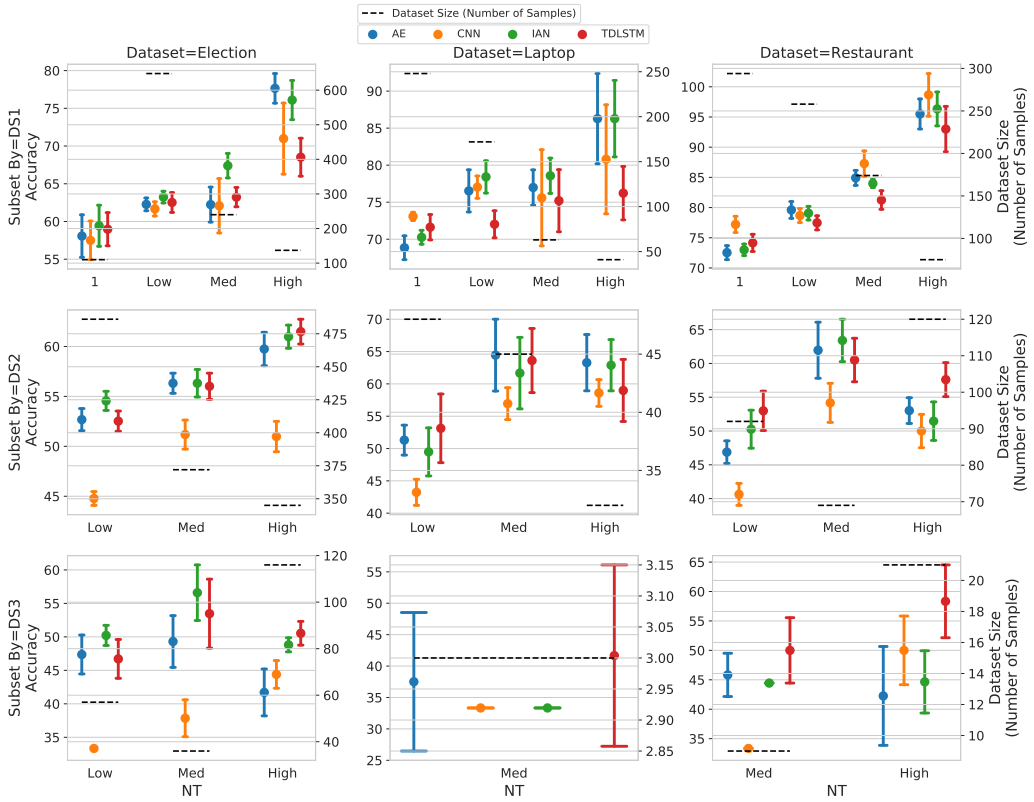


Figure B.3: Each plot shows the performance (y-axis accuracy) of the given models and sample size of the data evaluated on (y-axis dataset size) on the different validation datasets (columns) after being subsetted by the relevant *DS* subset (rows) and then NT subset (x-axis).

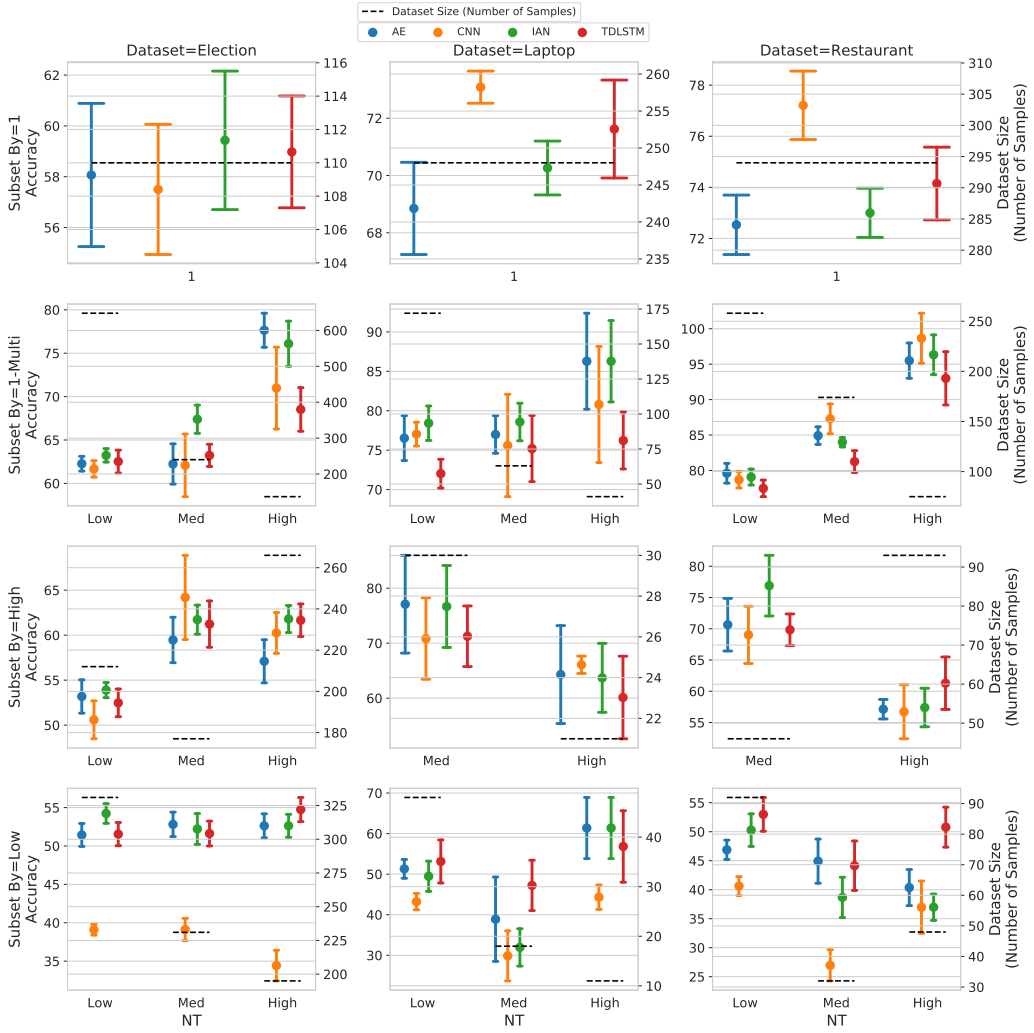


Figure B.4: Each plot shows the performance (y-axis accuracy) of the given models and sample size of the data evaluated on (y-axis dataset size) on the different validation datasets (columns) after being subsetted by the relevant *TSSR* subset (rows) and then NT subset (x-axis).

B.3 Text Classifier Performance[1]

The results of the *CNN* method trained using two different ways of creating a text level dataset from the TDSA datasets are reported. To re-iterate the two ways of creating a text level dataset are; 1. only use texts that contain one unique sentiment (single), and 2. use the majority sentiment from the text (average). The model trained using the first dataset creation will be called *CNN (single)* and the model trained using the second dataset creation will be called (*CNN (average)*). The training dataset statistics for these two datasets are shown in table B.7, from this it is clear and expected that the average dataset has far more samples. For clarification the validation and test split statistics are those of the TDSA dataset statistics found in table 4.13, as the text classification models are being tested as TDSA classifiers.

		Dataset		
	Creation Method	Election	Laptop	Restaurant
Sample Size	average	2319	1051	1378
	single	1227	933	1162
Negative	average	49%	43%	24%
	single	52%	44%	21%
Neutral	average	37%	17%	16%
	single	37%	14%	14%
Positive	average	15%	40%	60%
	single	11%	42%	64%

Table B.7: Dataset statistics for the training split for the two *CNN* models *average* and *single*. The Negative, Neutral, and Positive rows show the proportion of samples that represent the respective sentiment classes.

Tables B.8 and B.9 show the mean and standard deviation of the scores over eight runs on the validation and test splits respectively.

		CNN Model		
Metric	Dataset	*average*	*single*	Difference
Accuracy	Election	54.07 (0.56)	**54.54** (0.43)	-0.48 (0.67)
	Laptop	**70.65** (0.68)	69.46 (0.72)	1.19 (1.14)
	Restaurant	**72.31** (0.69)	71.98 (0.41)	0.34 (0.74)
Macro F1	Election	**42.74** (2.09)	39.62 (1.75)	3.12 (2.58)
	Laptop	**66.32** (0.96)	63.33 (1.70)	2.99 (2.18)
	Restaurant	**60.51** (1.20)	58.74 (1.44)	1.77 (1.69)

Table B.8: Validation results for *CNN (single)* and *CNN (average)*.

From these results the majority of the time *CNN (average)* is the better model, of which the result is larger on the Macro F1 metric. This is most likely due to the fact that the dataset for *CNN (single)* contains very few samples for the minority classes. Table

[1]Notebook that created these results can be found here: `https://github.com/apmoore1/tdsa_comparisons/blob/master/analysis/Baseline_non_target_results.ipynb`.

		CNN Model		
Metric	Dataset	*average*	*single*	Difference
Accuracy	Election	52.35 (0.69)	**54.29** (0.73)	-1.94 (1.36)
	Laptop	**68.26** (0.69)	65.99 (0.80)	2.27 (1.00)
	Restaurant	**75.81** (0.55)	75.19 (0.94)	0.62 (1.31)
Macro F1	Election	**39.98** (2.20)	39.73 (1.88)	0.24 (3.00)
	Laptop	**60.43** (1.36)	55.36 (2.00)	5.07 (2.76)
	Restaurant	**59.40** (1.52)	56.71 (1.63)	2.69 (2.82)

Table B.9: Test results for *CNN (single)* and *CNN (average)*

B.10 shows the p-values from the appropriate one-tailed hypothesis tests where the null hypothesis is that the *CNN (average)* model performs just as well as the *CNN (single)* model. After correcting the p-values using Bonferroni the *CNN (average)* is significantly better with a confidence of 95% on the validation split for 1 out of the 3 and 3 out of the 3 datasets for the accuracy and Macro F1 metrics respectively. On the test split after correcting the p-values using Bonferroni the *CNN (average)* is significantly better with a confidence of 95% for 1 out of the 3 and 2 out of the 3 datasets for the accuracy and Macro F1 metrics respectively.

		Dataset		
Split	Metric	Election	Laptop	Restaurant
Test	Accuracy	0.999921	$0.000029^{\dagger}$	0.077711
	Macro F1	0.444319	$0.005859^{\dagger}$	0.024975^{*}
Validation	Accuracy	0.950955	$0.003293^{\dagger}$	0.144942
	Macro F1	$0.005859^{\dagger}$	$0.005859^{\dagger}$	0.017846^{*}

Table B.10: P-Values. † and $*$ indicates p-values less than or equal to 0.01 and 0.05 respectively

Appendix C

Case Studies in Improving Experimental Methodology for TDSA

C.1 Figures

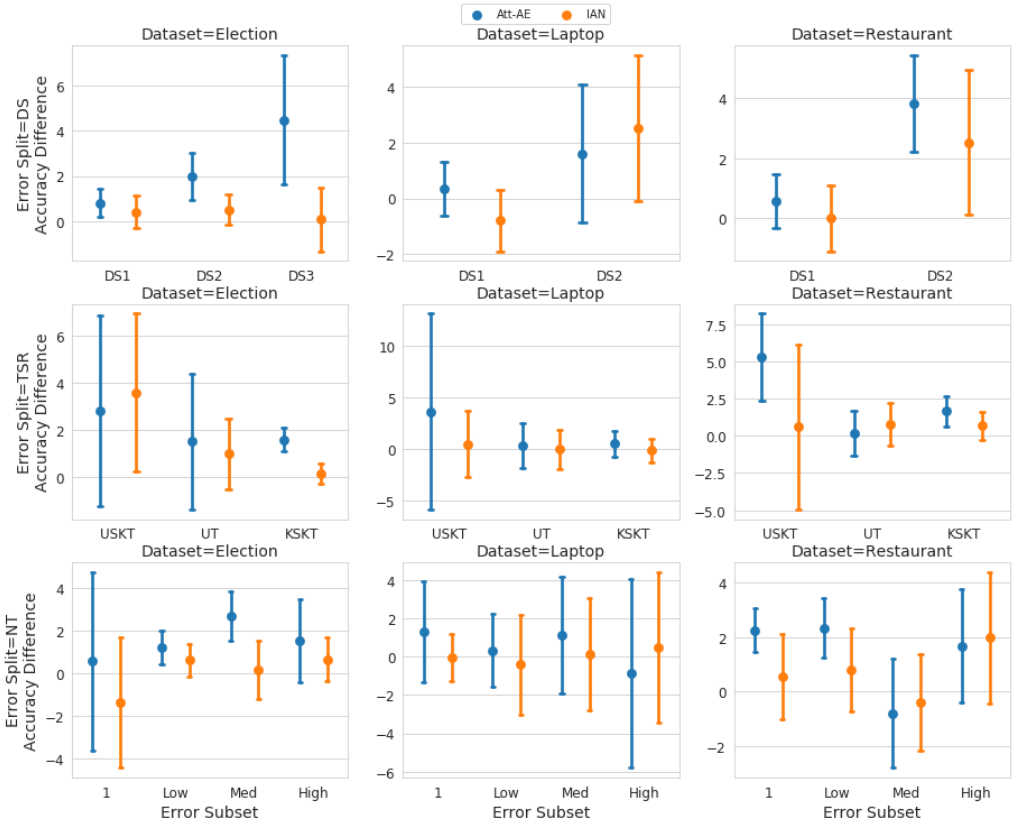


Figure C.1: Validation split results. Columns represent different datasets, rows different error splits. Each plot represents the differences between the position and baseline models for the Accuracy metric on the given error subset.

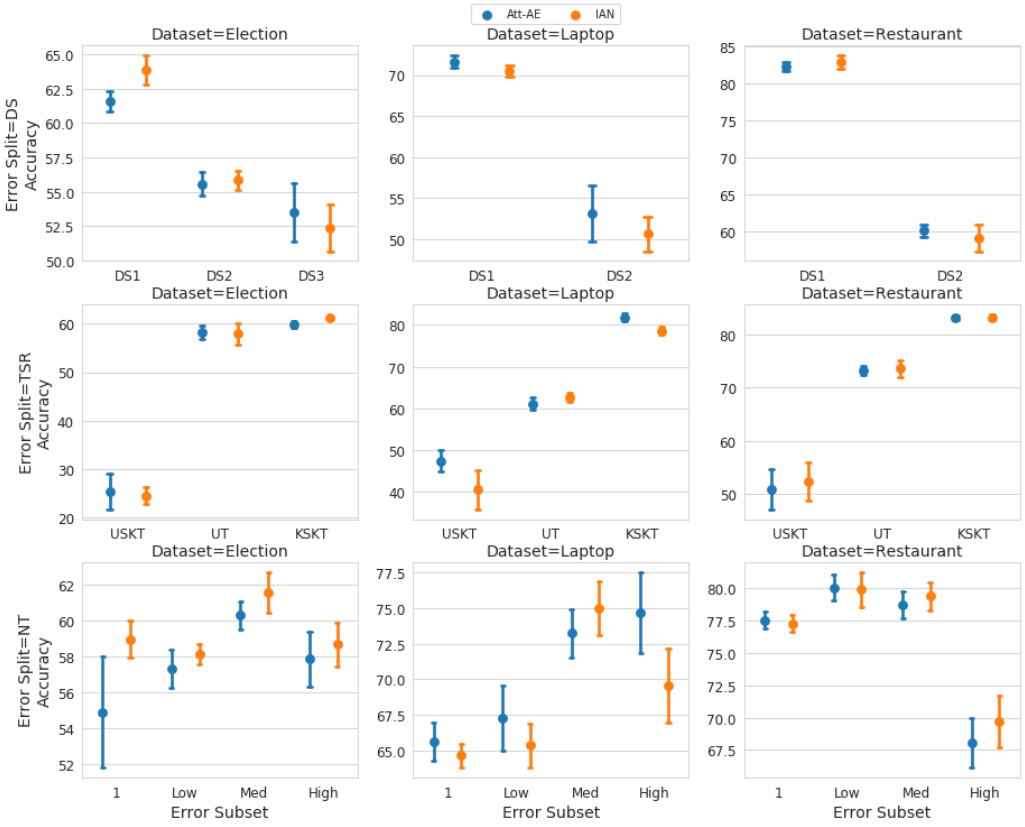


Figure C.2: Test split results. Columns represent different datasets, rows different error splits. Each plot represents the Accuracy metric for the position models on the given error subset.

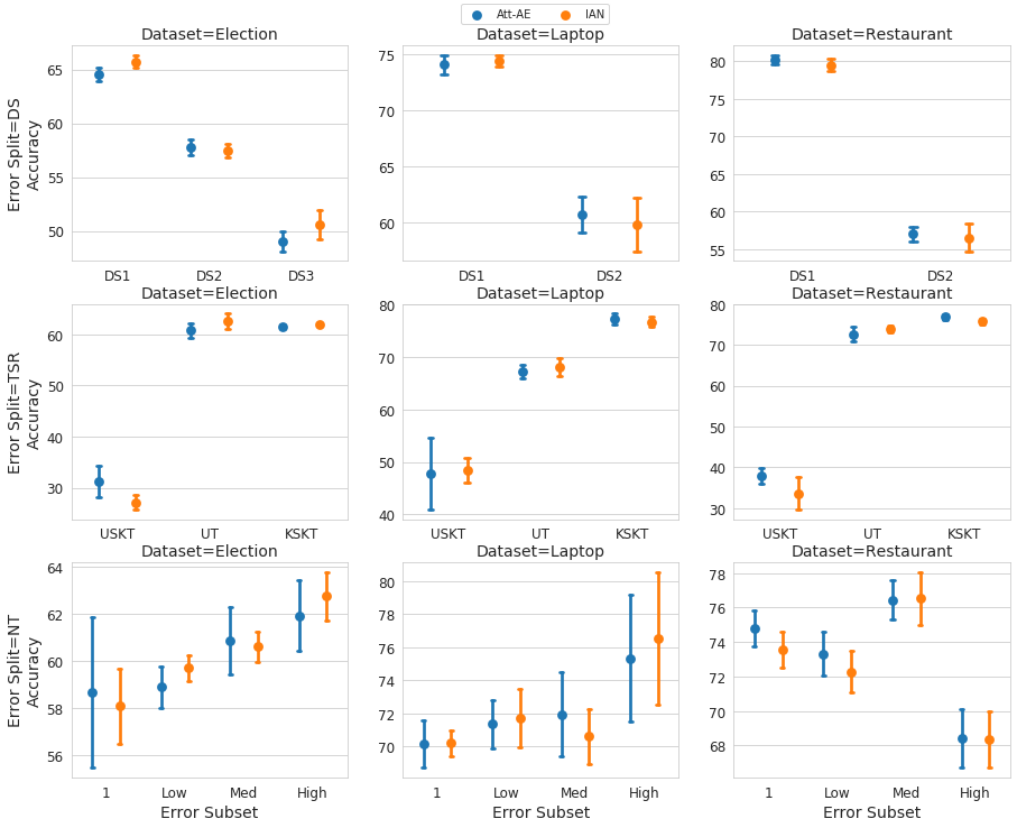


Figure C.3: Validation split results. Columns represent different datasets, rows different error splits. Each plot represents the Accuracy metric for the position models on the given error subset.

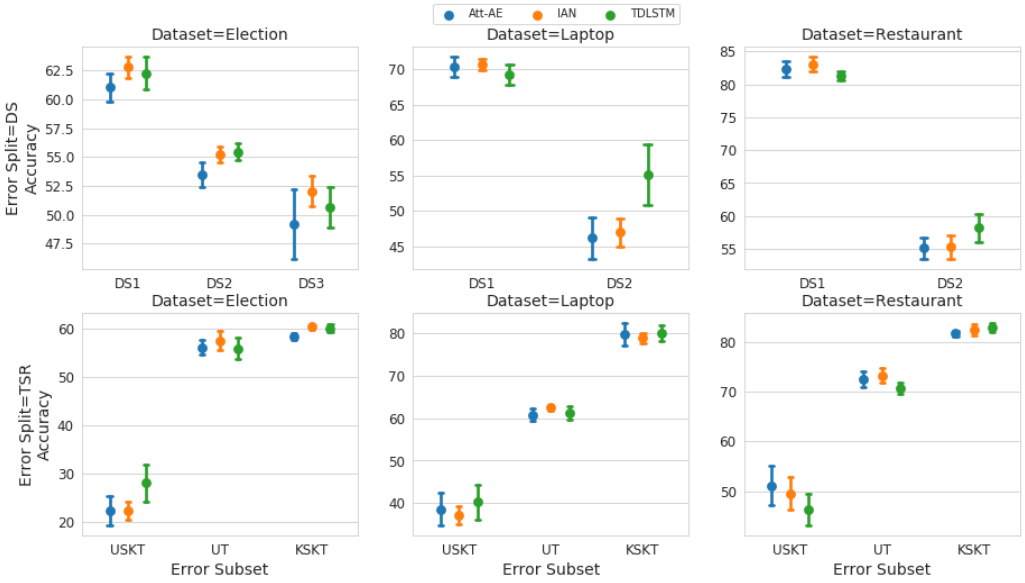


Figure C.4: Test split results. Columns represent different datasets, rows different error splits. Each plot represents the Accuracy metric for the target aware models on the given error subset.

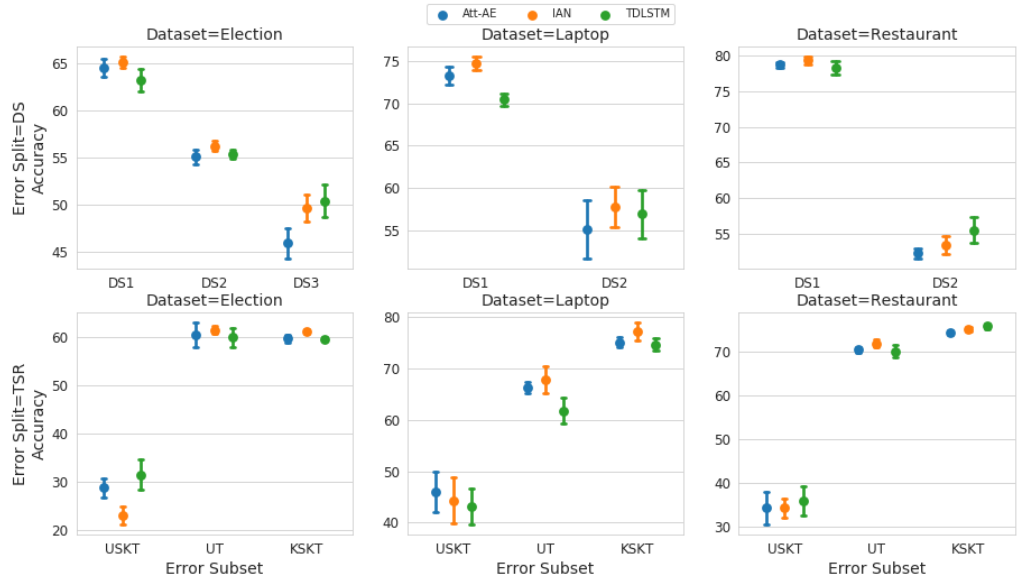


Figure C.5: Validation split results. Columns represent different datasets, rows different error splits. Each plot represents the Accuracy metric for the target aware models on the given error subset.

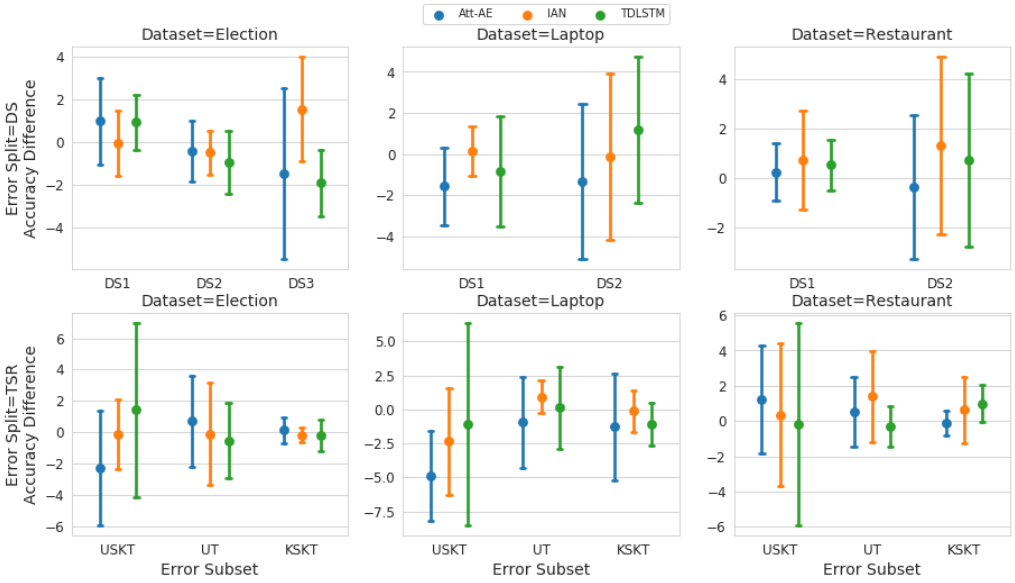


Figure C.6: Test split results. Columns represent different datasets, rows different error splits. Each plot represents the differences between the position and baseline models for the Accuracy metric on the given error subset.

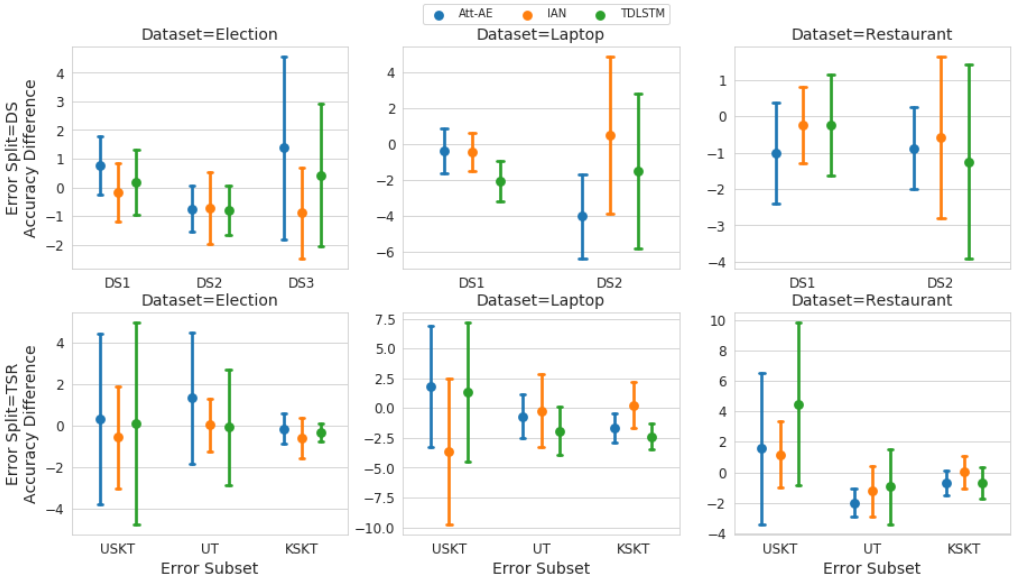


Figure C.7: Validation split results. Columns represent different datasets, rows different error splits. Each plot represents the differences between the target aware and baseline models for the Accuracy metric on the given error subset.

References

Abdou, Mostafa et al. (Nov. 2019). "X-WikiRE: A Large, Multilingual Resource for Relation Extraction as Machine Comprehension". In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 265–274. URL: https://www.aclweb.org/anthology/D19-6130.

Angelidis, Stefanos and Mirella Lapata (2018). "Multiple Instance Learning Networks for Fine-Grained Sentiment Analysis". In: *Transactions of the Association for Computational Linguistics* 6, pp. 17–31. URL: https://www.aclweb.org/anthology/Q18-1002.

Augenstein, Isabelle, Sebastian Ruder, and Anders Søgaard (June 2018). "Multi-Task Learning of Pairwise Sequence Classification Tasks over Disparate Label Spaces". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1896–1906. URL: https://www.aclweb.org/anthology/N18-1172.

Bao, Lingxian, Patrik Lambert, and Toni Badia (July 2019). "Attention and Lexicon Regularized LSTM for Aspect-based Sentiment Analysis". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Florence, Italy: Association for Computational Linguistics, pp. 253–259. URL: https://www.aclweb.org/anthology/P19-2035.

Barnes, Jeremy, Roman Klinger, and Sabine Schulte im Walde (Sept. 2017). "Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment Datasets". In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2–12. URL: https://www.aclweb.org/anthology/W17-5202.

Barnes, Jeremy, Lilja Øvrelid, and Erik Velldal (Aug. 2019). "Sentiment Analysis Is Not Solved! Assessing and Probing Sentiment Classification". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, pp. 12–23. URL: https://www.aclweb.org/anthology/W19-4802.

Barnes, Jeremy, Erik Velldal, and Lilja Øvrelid (2021). "Improving sentiment analysis with multi-task learning of negation". In: *Natural Language Engineering* 27.2, 249–269. URL: https://doi.org/10.1017/S1351324920000510.

Bayoudhi, Amine et al. (2015). "Sentiment Classification at Discourse Segment Level: Experiments on multi-domain Arabic corpus". In: *J. Lang. Technol. Comput. Linguistics* 30.1, pp. 1–24. URL: http://www.jlcl.org/2015_Heft1/1Bayoudhi.pdf.

Bender, Emily (2019). "The #BenderRule: On Naming the Languages We Study and Why It Matters". In: *The Gradient*. URL: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/.

Bender, Emily M (2011). "On achieving and evaluating language-independence in NLP". In: *Linguistic Issues in Language Technology* 6.3, pp. 1–26. URL: http://journals.linguisticsociety.org/elanguage/lilt/article/download/2624/2624-5403-1-PB.pdf.

Benjamini, Yoav and Ruth Heller (2008). "Screening for partial conjunction hypotheses". In: *Biometrics* 64.4, pp. 1215–1222. URL: https://doi.org/10.1111/j.1541-0420.2007.00984.x.

Berg-Kirkpatrick, Taylor, David Burkett, and Dan Klein (July 2012). "An Empirical Investigation of Statistical Significance in NLP". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pp. 995–1005. URL: https://www.aclweb.org/anthology/D12-1091.

Bethard, Steven et al. (June 2016). "SemEval-2016 Task 12: Clinical TempEval". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1052–1062. URL: https://www.aclweb.org/anthology/S16-1165.

Bhatia, Parminder, Yangfeng Ji, and Jacob Eisenstein (Sept. 2015). "Better Document-level Sentiment Analysis from RST Discourse Parsing". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 2212–2218. URL: https://www.aclweb.org/anthology/D15-1263.

Bojanowski, Piotr et al. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. URL: https://www.aclweb.org/anthology/Q17-1010.

Brahma, Siddhartha (2018). "Improved Sentence Modeling using Suffix Bidirectional LSTM". In: *arXiv preprint arXiv:1805.07340*. URL: https://arxiv.org/pdf/1805.07340.pdf.

Branco, António et al. (May 2020). "A Shared Task of a New, Collaborative Type to Foster Reproducibility: A First Exercise in the Area of Language Science and Technology with REPROLANG2020". English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 5539–5545. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.680.

Brants, Thorsten and Alex Franz (n.d.). "Web 1t 5-gram version 1 (2006)". In: *Linguistic Data Consortium, Philadelphia* (). URL: https://catalog.ldc.upenn.edu/LDC2006T13.

Brun, Caroline, Julien Perez, and Claude Roux (June 2016). "XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 277–281. URL: https://www.aclweb.org/anthology/S16-1044.

Camacho-Collados, Jose and Mohammad Taher Pilehvar (2018). "From word to sense embeddings: A survey on vector representations of meaning". In: *Journal of Artificial Intelligence Research* 63, pp. 743–788. URL: https://doi.org/10.1613/jair.1.11259.

Chang, Chih-Chung and Chih-Jen Lin (May 2011). "LIBSVM: A Library for Support Vector Machines". In: *ACM Trans. Intell. Syst. Technol.* 2.3. ISSN: 2157-6904. URL: https://doi.org/10.1145/1961189.1961199.

Chelba, Ciprian et al. (2014). "One billion word benchmark for measuring progress in statistical language modeling". In: *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. Ed. by Haizhou Li et al. ISCA, pp. 2635–2639. URL: http://www.isca-speech.org/archive/interspeech_2014/i14_2635.html.

Chen, Peng et al. (Sept. 2017). "Recurrent Attention Network on Memory for Aspect Sentiment Analysis". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 452–461. URL: https://www.aclweb.org/anthology/D17-1047.

Chen, Zhuang and Tieyun Qian (July 2019). "Transfer Capsule Network for Aspect Level Sentiment Classification". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 547–556. URL: https://www.aclweb.org/anthology/P19-1052.

Chernyshevich, Maryna (Aug. 2014). "IHS R&D Belarus: Cross-domain extraction of product features using CRF". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 309–313. URL: https://www.aclweb.org/anthology/S14-2051.

Cho, Kyunghyun et al. (Oct. 2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. URL: https://www.aclweb.org/anthology/D14-1179.

Choi, Yejin, Eric Breck, and Claire Cardie (July 2006). "Joint Extraction of Entities and Relations for Opinion Recognition". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, pp. 431–439. URL: https://www.aclweb.org/anthology/W06-1651.

Choi, Yejin and Claire Cardie (Oct. 2008). "Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 793–801. URL: https://www.aclweb.org/anthology/D08-1083.

Choi, Yejin et al. (Oct. 2005). "Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 355–362. URL: https://www.aclweb.org/anthology/H05-1045.

Chollet, François et al. (2015). *Keras*. https://keras.io.

Church, Kenneth Ward (Feb. 1988). "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text". In: *Second Conference on Applied Natural Language Processing*. Austin, Texas, USA: Association for Computational Linguistics, pp. 136–143. URL: https://www.aclweb.org/anthology/A88-1019.

Church, Kenneth Ward and Patrick Hanks (June 1989). "Word Association Norms, Mutual Information, and Lexicography". In: *27th Annual Meeting of the Association for Computational Linguistics*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 76–83. URL: https://www.aclweb.org/anthology/P89-1010.

Collberg, Christian and Todd A. Proebsting (Feb. 2016). "Repeatability in Computer Systems Research". In: *Commun. ACM* 59.3, pp. 62–69. ISSN: 0001-0782. URL: http://doi.acm.org/10.1145/2812803.

Collins, Michael (2003). "Head-Driven Statistical Models for Natural Language Parsing". In: *Computational Linguistics* 29.4, pp. 589–637. URL: https://www.aclweb.org/anthology/J03-4003.

Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, pp. 160–167. ISBN: 9781605582054. URL: https://doi.org/10.1145/1390156.1390177.

Conneau, Alexis and Guillaume Lample (2019). "Cross-lingual Language Model Pretraining". In: *Advances in Neural Information Processing Systems*, pp. 7057–7067. URL: https://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf.

Conneau, Alexis et al. (Apr. 2017). "Very Deep Convolutional Networks for Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 1107–1116. URL: https://www.aclweb.org/anthology/E17-1104.

Dai, Andrew M and Quoc V Le (2015). "Semi-supervised sequence learning". In: *Advances in neural information processing systems*, pp. 3079–3087. URL: http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning.pdf.

Dashtipour, Kia et al. (2016). "Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques". In: *Cogn. Comput.* 8.4, pp. 757–771. URL: https://doi.org/10.1007/s12559-016-9415-7.

Daumé III, Hal (June 2007). "Frustratingly Easy Domain Adaptation". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 256–263. URL: https://www.aclweb.org/anthology/P07-1033.

De Marneffe, Marie-Catherine and Christopher D Manning (2008). *Stanford typed dependencies manual*. Tech. rep. Technical report, Stanford University. URL: https://nlp.stanford.edu/static/software/dependencies_manual.pdf.

Deng, Lingjia, Yoonjung Choi, and Janyce Wiebe (Aug. 2013). "Benefactive/Malefactive Event and Writer Attitude Annotation". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia,

Bulgaria: Association for Computational Linguistics, pp. 120–125. URL: https://www.aclweb.org/anthology/P13-2022.

Deng, Lingjia and Janyce Wiebe (Apr. 2014a). "Sentiment Propagation via Implicature Constraints". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 377–385. URL: https://www.aclweb.org/anthology/E14-1040.

— (Sept. 2015a). "Joint Prediction for Entity/Event-Level Sentiment Analysis using Probabilistic Soft Logic Models". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 179–189. URL: https://www.aclweb.org/anthology/D15-1018.

— (2015b). "MPQA 3.0: An Entity/Event-Level Sentiment Corpus". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1323–1328. URL: https://www.aclweb.org/anthology/N15-1146.

Deng, Lingjia, Janyce Wiebe, and Yoonjung Choi (Aug. 2014b). "Joint Inference and Disambiguation of Implicit Sentiments via Implicature Constraints". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp. 79–88. URL: https://www.aclweb.org/anthology/C14-1009.

Devlin, Jacob et al. (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423.

Dietterich, Thomas G, Richard H Lathrop, and Tomás Lozano-Pérez (1997). "Solving the multiple instance problem with axis-parallel rectangles". In: *Artificial intelligence* 89.1-2, pp. 31–71. URL: https://doi.org/10.1016/S0004-3702(96)00034-3.

Ding, Haibo and Ellen Riloff (June 2018). "Human Needs Categorization of Affective Events Using Labeled and Unlabeled Data". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1919–1929. URL: https://www.aclweb.org/anthology/N18-1174.

Ding, Xiaowen and Bing Liu (Aug. 2010). "Resolving Object and Attribute Coreference in Opinion Mining". In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pp. 268–276. URL: https://www.aclweb.org/anthology/C10-1031.

Ding, Xiaowen, Bing Liu, and Philip S. Yu (2008). "A Holistic Lexicon-Based Approach to Opinion Mining". In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*. WSDM '08. Palo Alto, California, USA: Association for

Computing Machinery, 231–240. ISBN: 9781595939272. URL: https://doi.org/10.1145/1341531.1341561.

Dodge, Jesse et al. (Nov. 2019). "Show Your Work: Improved Reporting of Experimental Results". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2185–2194. URL: https://www.aclweb.org/anthology/D19-1224.

Dong, Li et al. (June 2014). "Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 49–54. URL: https://www.aclweb.org/anthology/P14-2009.

Dozat, Timothy and Christopher D. Manning (2017). "Deep Biaffine Attention for Neural Dependency Parsing". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=Hk95PK9le.

Dridan, Rebecca and Stephan Oepen (July 2012). "Tokenization: Returning to a Long Solved Problem — A Survey, Contrastive Experiment, Recommendations, and Toolkit —". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, pp. 378–382. URL: https://www.aclweb.org/anthology/P12-2074.

Dror, Rotem et al. (2017). "Replicability Analysis for Natural Language Processing: Testing Significance with Multiple Datasets". In: *Transactions of the Association for Computational Linguistics* 5, pp. 471–486. URL: https://www.aclweb.org/anthology/Q17-1033.

Dror, Rotem et al. (July 2018). "The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1383–1392. URL: https://www.aclweb.org/anthology/P18-1128.

Du, Chunning et al. (Nov. 2019). "Capsule Network with Interactive Attention for Aspect-Level Sentiment Classification". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5489–5498. URL: https://www.aclweb.org/anthology/D19-1551.

Du Bois, John W (2007). "The stance triangle". In: *Pragmatics & beyond. New series* 164, pp. 139–182. URL: https://doi.org/10.1075/pbns.164.07du.

Efron, Bradley and Robert Tibshirani (1994). *An introduction to the bootstrap*. Chapman and Hall. ISBN: 978-0412042317.

El-Haj, Mahmoud et al. (May 2016). "Learning Tone and Attribution for Financial Text Mining". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 1820–1825. URL: https://www.aclweb.org/anthology/L16-1287.

El-Haj, Mahmoud et al. (2019). "In search of meaning: Lessons, resources and next steps for computational analysis of financial discourse". In: *Journal of Business Finance & Accounting* 46.3-4, pp. 265–306. URL: https://doi.org/10.1111/jbfa.12378.

Fan, Feifan, Yansong Feng, and Dongyan Zhao (2018). "Multi-grained Attention Network for Aspect-Level Sentiment Classification". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3433–3442. URL: https://www.aclweb.org/anthology/D18-1380.

Fan, Rong-En et al. (2008). "LIBLINEAR: A library for large linear classification". In: *Journal of machine learning research* 9.Aug, pp. 1871–1874. URL: http://www.jmlr.org/papers/volume9/fan08a/fan08a.pdf.

Ferro, Nicola and Donna Harman (2009). "CLEF 2009: Grid@CLEF Pilot Track Overview". In: *Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*. Ed. by Carol Peters et al. Vol. 6241. Lecture Notes in Computer Science. Springer, pp. 552–565. URL: https://doi.org/10.1007/978-3-642-15754-7_68.

Fokkens, Antske et al. (Aug. 2013). "Offspring from Reproduction Problems: What Replication Failure Teaches Us". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1691–1701. URL: https://www.aclweb.org/anthology/P13-1166.

Gal, Yarin and Zoubin Ghahramani (2016). "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al., pp. 1019–1027. URL: http://papers.nips.cc/paper/6241-a-theoretically-grounded-application-of-dropout-in-recurrent-neural-networks.

Ganganwar, Vaishali and R. Rajalakshmi (2019). "Implicit Aspect Extraction for Sentiment Analysis: A Survey of Recent Approaches". In: *Procedia Computer Science* 165. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019, pp. 485 –491. ISSN: 1877-0509. URL: https://doi.org/10.1016/j.procs.2020.01.010.

Ganu, Gayatree, Noémie Elhadad, and Amélie Marian (2009). "Beyond the Stars: Improving Rating Predictions using Review Text Content". In: *12th International Workshop on the Web and Databases, WebDB 2009, Providence, Rhode Island, USA, June 28, 2009*. URL: http://webdb09.cse.buffalo.edu/papers/Paper9/WebDB.pdf.

Gardner, Matt et al. (July 2018). "AllenNLP: A Deep Semantic Natural Language Processing Platform". In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1–6. URL: https://www.aclweb.org/anthology/W18-2501.

Gimpel, Kevin et al. (June 2011). "Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 42–47. URL: https://www.aclweb.org/anthology/P11-2008.

Go, Alec, Richa Bhayani, and Lei Huang (2009). *Twitter sentiment classification using distant supervision*. CS224N Project Report. Stanford. URL: https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

Goldberg, Yoav (2017). "Neural network methods for natural language processing". In: *Synthesis Lectures on Human Language Technologies* 10.1, pp. 1–309. URL: https://doi.org/10.2200/S00762ED1V01Y201703HLT037.

Gorman, Kyle and Steven Bedrick (July 2019). "We Need to Talk about Standard Splits". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2786–2791. URL: https://www.aclweb.org/anthology/P19-1267.

Gu, Shuqin et al. (Aug. 2018). "A Position-aware Bidirectional Attention Network for Aspect-level Sentiment Analysis". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 774–784. URL: https://www.aclweb.org/anthology/C18-1066.

Gururangan, Suchin et al. (2020). "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". In: *arXiv preprint arXiv:2004.10964*. URL: https://arxiv.org/pdf/2004.10964.pdf.

Han, Wen-Bin and Noriko Kando (June 2019). "Opinion Mining with Deep Contextualized Embeddings". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 35–42. URL: https://www.aclweb.org/anthology/N19-3006.

Hashimoto, Kazuma et al. (Sept. 2017). "A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1923–1933. URL: https://www.aclweb.org/anthology/D17-1206.

Hazarika, Devamanyu et al. (June 2018). "Modeling Inter-Aspect Dependencies for Aspect-Based Sentiment Analysis". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 266–270. URL: https://www.aclweb.org/anthology/N18-2043.

He, Ruidan et al. (Aug. 2018a). "Effective Attention Modeling for Aspect-Level Sentiment Classification". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1121–1131. URL: https://www.aclweb.org/anthology/C18-1096.

— (July 2018b). "Exploiting Document Knowledge for Aspect-level Sentiment Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 579–585. URL: https://www.aclweb.org/anthology/P18-2092.

He, Ruining and Julian McAuley (2016). "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering". In: *proceedings of the 25th international conference on world wide web*. International World Wide Web Confer-

ences Steering Committee, pp. 507–517. URL: https://doi.org/10.1145/2872427.2883037.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Howard, Jeremy and Sebastian Ruder (July 2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 328–339. URL: https://www.aclweb.org/anthology/P18-1031.

Hsu, Chih-Wei, Chih-Chung Chang, Chih-Jen Lin, et al. (2016). "A practical guide to support vector classification". In: URL: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Hu, Mengting et al. (Nov. 2019a). "CAN: Constrained Attention Networks for Multi-Aspect Sentiment Analysis". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 4601–4610. URL: https://www.aclweb.org/anthology/D19-1467.

Hu, Minghao et al. (July 2019b). "Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 537–546. URL: https://www.aclweb.org/anthology/P19-1051.

Hu, Minqing and Bing Liu (2004a). "Mining and Summarizing Customer Reviews". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: Association for Computing Machinery, 168–177. ISBN: 1581138881. URL: https://doi.org/10.1145/1014052.1014073.

— (2004b). "Mining Opinion Features in Customer Reviews". In: *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*. Ed. by Deborah L. McGuinness and George Ferguson. AAAI Press / The MIT Press, pp. 755–760. URL: http://www.aaai.org/Library/AAAI/2004/aaai04-119.php.

Huang, Binxuan and Kathleen Carley (Nov. 2019). "Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5469–5477. URL: https://www.aclweb.org/anthology/D19-1549.

Hummel, Robert A and Steven W Zucker (1983). "On the foundations of relaxation labeling processes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3, pp. 267–287.

Hutto, Clayton J. and Eric Gilbert (2014). "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *ICWSM*. URL: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109/8122.

Irsoy, Ozan and Claire Cardie (2014). "Deep recursive neural networks for compositionality in language". In: *Advances in neural information processing systems*, pp. 2096–2104. URL: http://papers.nips.cc/paper/5551-deep-recursive-neural-networks-for-compositionality-in-language.pdf.

İrsoy, Ozan and Claire Cardie (Oct. 2014). "Opinion Mining with Deep Recurrent Neural Networks". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 720–728. URL: https://www.aclweb.org/anthology/D14-1080.

Jakob, Niklas and Iryna Gurevych (Oct. 2010a). "Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, pp. 1035–1045. URL: https://www.aclweb.org/anthology/D10-1101.

— (July 2010b). "Using Anaphora Resolution to Improve Opinion Target Identification in Movie Reviews". In: *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, pp. 263–268. URL: https://www.aclweb.org/anthology/P10-2049.

Jebbara, Soufian and Philipp Cimiano (2016). "Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture". In: *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. Ed. by Gal A. Kaminka et al. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 1123–1131. URL: https://doi.org/10.3233/978-1-61499-672-9-1123.

— (Sept. 2017). "Improving Opinion-Target Extraction with Character-Level Word Embeddings". In: *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 159–167. URL: https://www.aclweb.org/anthology/W17-4124.

— (June 2019). "Zero-Shot Cross-Lingual Opinion Target Extraction". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2486–2495. URL: https://www.aclweb.org/anthology/N19-1257.

Jiang, Qingnan et al. (Nov. 2019). "A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 6280–6285. URL: https://www.aclweb.org/anthology/D19-1654.

Jin, Wei, Hung Hay Ho, and Rohini K. Srihari (2009). "OpinionMiner: A Novel Machine Learning System for Web Opinion Mining and Extraction". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining. KDD '09. Paris, France: Association for Computing Machinery, 1195–1204. ISBN: 9781605584959. URL: https://doi.org/10.1145/1557019.1557148.

Johansson, Richard and Alessandro Moschitti (Aug. 2010). "Reranking Models in Fine-grained Opinion Analysis". In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pp. 519–527. URL: https://www.aclweb.org/anthology/C10-1059.

Johnson, Rie and Tong Zhang (2015). "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 103–112. URL: https://www.aclweb.org/anthology/N15-1011.

Jones, Karen Sparck (1972). "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of documentation*. URL: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=F269941CD77272DD5A365D4C3562A3A0?doi=10.1.1.115.8343&rep=rep1&type=pdf.

Joshi, Vidur, Matthew Peters, and Mark Hopkins (July 2018). "Extending a Parser to Distant Domains Using a Few Dozen Partially Annotated Examples". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1190–1199. URL: https://www.aclweb.org/anthology/P18-1110.

Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom (June 2014). "A Convolutional Neural Network for Modelling Sentences". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 655–665. URL: https://www.aclweb.org/anthology/P14-1062.

Kaljahi, Rasoul and Jennifer Foster (Oct. 2018). "Sentiment Expression Boundaries in Sentiment Polarity Classification". In: *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, pp. 156–166. URL: https://www.aclweb.org/anthology/W18-6222.

Kaplan, Ronald M (2005). "A method for tokenizing text". In: *Inquiries into words, constraints and contexts* 55. URL: http://web.stanford.edu/group/cslipublications/cslipublications/koskenniemi-festschrift/kk-festschrift-all-2005.pdf#page=79.

Karamanolakis, Giannis, Daniel Hsu, and Luis Gravano (Nov. 2019). "Weakly Supervised Attention Networks for Fine-Grained Opinion Mining and Public Health". In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 1–10. URL: https://www.aclweb.org/anthology/D19-5501.

Karpathy, Andrej and Li Fei-Fei (2015). "Deep Visual-Semantic Alignments for Generating Image Descriptions". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Karpathy_Deep_Visual-Semantic_Alignments_2015_CVPR_paper.pdf.

Katiyar, Arzoo and Claire Cardie (Aug. 2016). "Investigating LSTMs for Joint Extraction of Opinion Entities and Relations". In: *Proceedings of the 54th Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, pp. 919–929. URL: https://www.aclweb.org/anthology/P16-1087.

Kauter, Marjan Van de, Bart Desmet, and Véronique Hoste (2015). "The good, the bad and the implicit: a comprehensive approach to annotating explicit and implicit sentiment". In: *Language resources and evaluation* 49.3, pp. 685–720.

Kessler, Jason S and Nicolas Nicolov (2009). "Targeting sentiment expressions through supervised ranking of linguistic configurations". In: *Third International AAAI Conference on Weblogs and Social Media*. URL: https://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/190/413.

Kessler, Jason S. et al. (2010). "The ICWSM 2010 JDPA Sentiment Corpus for the Automotive". In: *4th International AAAI Conference on Weblogs and Social Media Data Challenge Workshop*. URL: https://www.icwsm.org/2010/papers/icwsm10dcw_8.pdf.

Kim, Soo-Min and Eduard Hovy (2004). "Determining the Sentiment of Opinions". In: *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. Geneva, Switzerland: COLING, pp. 1367–1373. URL: https://www.aclweb.org/anthology/C04-1200.

Kim, Yoon (Oct. 2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1746–1751. URL: https://www.aclweb.org/anthology/D14-1181.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980.

Kiritchenko, Svetlana et al. (Aug. 2014). "NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 437–442. URL: https://www.aclweb.org/anthology/S14-2076.

Koehn, Philipp (July 2004). "Statistical Significance Tests for Machine Translation Evaluation". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, pp. 388–395. URL: https://www.aclweb.org/anthology/W04-3250.

Kong, Lingpeng et al. (Oct. 2014). "A Dependency Parser for Tweets". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1001–1012. URL: https://www.aclweb.org/anthology/D14-1108.

Krippendorff, Klaus (2018). *Content analysis: An introduction to its methodology*. Sage publications. ISBN: 9781506395661.

Küçük, Dilek and Fazli Can (2020). "Stance Detection: A Survey". In: *ACM Computing Surveys (CSUR)* 53.1, pp. 1–37.

Kumar, Avinash et al. (2020). "Aspect Based Sentiment Classification Using Interactive Gated Convolutional Network". In: *IEEE Access*. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8972363.

Kumar, Ayush et al. (June 2016). "IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1129–1135. URL: https://www.aclweb.org/anthology/S16-1174.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*. Ed. by Carla E. Brodley and Andrea Pohoreckyj Danyluk. Morgan Kaufmann, pp. 282–289. URL: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers.

Le, Quoc and Tomas Mikolov (2014). "Distributed representations of sentences and documents". In: *International conference on machine learning*, pp. 1188–1196. URL: http://proceedings.mlr.press/v32/le14.pdf.

Lei, Tao, Regina Barzilay, and Tommi Jaakkola (Nov. 2016). "Rationalizing Neural Predictions". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 107–117. URL: https://www.aclweb.org/anthology/D16-1011.

Levy, Omer et al. (Aug. 2017). "Zero-Shot Relation Extraction via Reading Comprehension". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 333–342. URL: https://www.aclweb.org/anthology/K17-1034.

Li, Junjie, Haitong Yang, and Chengqing Zong (Aug. 2018a). "Document-level Multi-aspect Sentiment Classification by Jointly Modeling Users, Aspects, and Overall Ratings". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 925–936. URL: https://www.aclweb.org/anthology/C18-1079.

Li, Lishuang, Yang Liu, and AnQiao Zhou (Oct. 2018b). "Hierarchical Attention Based Position-Aware Network for Aspect-Level Sentiment Analysis". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, pp. 181–189. URL: https://www.aclweb.org/anthology/K18-1018.

Li, Xin and Wai Lam (Sept. 2017). "Deep Multi-Task Learning for Aspect Term Extraction with Memory Interaction". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2886–2892. URL: https://www.aclweb.org/anthology/D17-1310.

Li, Xin et al. (2018c). "Aspect Term Extraction with History Attention and Selective Transformation". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 4194–4200. ISBN: 9780999241127. URL: https://www.ijcai.org/Proceedings/2018/0583.pdf.

Li, Xin et al. (July 2018d). "Transformation Networks for Target-Oriented Sentiment Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Associa-

tion for Computational Linguistics, pp. 946–956. URL: https://www.aclweb.org/anthology/P18-1087.

Li, Yingjie and Cornelia Caragea (Nov. 2019). "Multi-Task Stance Detection with Sentiment and Stance Lexicons". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 6299–6305. URL: https://www.aclweb.org/anthology/D19-1657.

Ling, Wang et al. (Sept. 2015). "Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1520–1530. URL: https://www.aclweb.org/anthology/D15-1176.

Liu, Bing (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press. ISBN: 9781107017894.

Liu, Jiangming and Yue Zhang (Apr. 2017). "Attention Modeling for Targeted Sentiment". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 572–577. URL: https://www.aclweb.org/anthology/E17-2091.

Liu, Kang, Liheng Xu, and Jun Zhao (July 2012). "Opinion Target Extraction Using Word-Based Translation Model". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pp. 1346–1356. URL: https://www.aclweb.org/anthology/D12-1123.

Liu, Nelson F. et al. (June 2019a). "Linguistic Knowledge and Transferability of Contextual Representations". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1073–1094. URL: https://www.aclweb.org/anthology/N19-1112.

Liu, Pengfei, Shafiq Joty, and Helen Meng (Sept. 2015). "Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1433–1443. URL: https://www.aclweb.org/anthology/D15-1168.

Liu, Qiao et al. (2018). "Content attention model for aspect based sentiment analysis". In: *Proceedings of the 2018 World Wide Web Conference*, pp. 1023–1032.

Liu, Xiaodong et al. (July 2019b). "Multi-Task Deep Neural Networks for Natural Language Understanding". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4487–4496.

Liu, Yinhan et al. (2019c). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692*. URL: https://arxiv.org/pdf/1907.11692.pdf.

Louridas, Panos and Georgios Gousios (Sept. 2012). "A Note on Rigour and Replicability". In: *SIGSOFT Softw. Eng. Notes* 37.5, 1–4. ISSN: 0163-5948. URL: https://doi.org/10.1145/2347696.2347706.

Luong, Thang, Hieu Pham, and Christopher D. Manning (Sept. 2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. URL: https://www.aclweb.org/anthology/D15-1166.

Ma, Dehong et al. (2017). "Interactive attention networks for aspect-level sentiment classification". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, pp. 4068–4074. URL: https://www.ijcai.org/proceedings/2017/0568.pdf.

Maas, Andrew L. et al. (June 2011). "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 142–150. URL: https://www.aclweb.org/anthology/P11-1015.

Mæhlum, Petter et al. (2019). "Annotating evaluative sentences for sentiment analysis: a dataset for Norwegian". In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. Turku, Finland: Linköping University Electronic Press, pp. 121–130. URL: https://www.aclweb.org/anthology/W19-6113.

Majumder, Navonil et al. (2018). "IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3402–3411. URL: https://www.aclweb.org/anthology/D18-1377.

Mann, William C. (July 1984). "Discourse Structures for Text Generation". In: *10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*. Stanford, California, USA: Association for Computational Linguistics, pp. 367–375. URL: https://www.aclweb.org/anthology/P84-1076.

Manning, Christopher et al. (June 2014). "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, pp. 55–60. URL: https://www.aclweb.org/anthology/P14-5010.

Marasovic, Ana (2020). "Deep Learning With Sentiment Inference For Discourse-Oriented Opinion Analysis". PhD thesis.

Marasović, Ana and Anette Frank (June 2018). "SRL4ORL: Improving Opinion Role Labeling Using Multi-Task Learning with Semantic Role Labeling". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 583–594. URL: https://www.aclweb.org/anthology/N18-1054.

Marrese-Taylor, Edison, Jorge Balazs, and Yutaka Matsuo (Sept. 2017a). "Mining fine-grained opinions on closed captions of YouTube videos with an attention-RNN".

In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 102–111. URL: https://www.aclweb.org/anthology/W17-5213.

Marrese-Taylor, Edison and Yutaka Matsuo (Apr. 2017b). "Replication issues in syntax-based aspect extraction for opinion mining". In: *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, pp. 23–32. URL: https://www.aclweb.org/anthology/E17-4003.

Martineau, Justin Christopher and Tim Finin (2009). "Delta tfidf: An improved feature space for sentiment analysis". In: *Third international AAAI conference on weblogs and social media*. URL: https://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/187/504.

McAuley, Julian et al. (2015). "Image-based recommendations on styles and substitutes". In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52. URL: https://bit.ly/39oDqK7.

McAuley, Julian J., Jure Leskovec, and Dan Jurafsky (2012). "Learning Attitudes and Attributes from Multi-aspect Reviews". In: *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*. Ed. by Mohammed Javeed Zaki et al. IEEE Computer Society, pp. 1020–1025. URL: https://doi.org/10.1109/ICDM.2012.110.

McDonald, Ryan et al. (June 2007). "Structured Models for Fine-to-Coarse Sentiment Analysis". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 432–439. URL: https://www.aclweb.org/anthology/P07-1055.

Mikolov, Tomas et al. (2013a). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119. URL: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Mikolov, Tomas et al. (2013b). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*. URL: https://arxiv.org/abs/1301.3781.

Miller, George A. (Nov. 1995). "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11, 39–41. ISSN: 0001-0782. URL: https://doi.org/10.1145/219717.219748.

Mitchell, Margaret et al. (Oct. 2013). "Open Domain Targeted Sentiment". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1643–1654. URL: https://www.aclweb.org/anthology/D13-1171.

Mitchell, Margaret et al. (2019). "Model Cards for Model Reporting". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. FAT* '19. Atlanta, GA, USA: Association for Computing Machinery, 220–229. ISBN: 9781450361255. URL: https://doi.org/10.1145/3287560.3287596.

Mohammad, Saif and Peter Turney (June 2010). "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon". In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and*

Generation of Emotion in Text. Los Angeles, CA: Association for Computational Linguistics, pp. 26–34. URL: https://www.aclweb.org/anthology/W10-0204.

Mohammad, Saif et al. (June 2016). "SemEval-2016 Task 6: Detecting Stance in Tweets". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 31–41. URL: https://www.aclweb.org/anthology/S16-1003.

Mohammad, Saif M., Parinaz Sobhani, and Svetlana Kiritchenko (June 2017). "Stance and Sentiment in Tweets". In: *ACM Trans. Internet Technol.* 17.3. ISSN: 1533-5399. URL: https://doi.org/10.1145/3003433.

Moore, Andrew and Paul Rayson (Aug. 2017). "Lancaster A at SemEval-2017 Task 5: Evaluation metrics matter: predicting sentiment from financial news headlines". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 581–585. URL: https://www.aclweb.org/anthology/S17-2095.

— (Aug. 2018). "Bringing replication and reproduction together with generalisability in NLP: Three reproduction studies for Target Dependent Sentiment Analysis". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1132–1144. URL: https://www.aclweb.org/anthology/C18-1097.

Moore, Andrew, Paul Edward Rayson, and Steven Eric Young (2016). "Domain adaptation using stock market prices to refine sentiment dictionaries". In: *Proceedings of the Emotion and Sentiment Analysis Workshop LREC 2016, Portorož, Slovenia*, pp. 63–66. URL: http://gsi.dit.upm.es/esa2016/Proceedings-ESA2016.pdf.

Moss, Henry et al. (July 2019). "FIESTA: Fast IdEntification of State-of-The-Art models using adaptive bandit algorithms". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2920–2930. URL: https://www.aclweb.org/anthology/P19-1281.

Mullen, Tony and Nigel Collier (July 2004). "Sentiment Analysis using Support Vector Machines with Diverse Information Sources". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, pp. 412–418. URL: https://www.aclweb.org/anthology/W04-3253.

Nakagawa, Tetsuji, Kentaro Inui, and Sadao Kurohashi (June 2010). "Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 786–794. URL: https://www.aclweb.org/anthology/N10-1120.

Nakov, Preslav et al. (June 2016). "SemEval-2016 Task 4: Sentiment Analysis in Twitter". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1–18. URL: https://www.aclweb.org/anthology/S16-1001.

Nangia, Nikita et al. (Sept. 2017). "The RepEval 2017 Shared Task: Multi-Genre Natural Language Inference with Sentence Representations". In: *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Copenhagen, Denmark:

Association for Computational Linguistics, pp. 1–10. URL: https://www.aclweb.org/anthology/W17-5301.

Nasukawa, Tetsuya and Jeonghee Yi (2003). "Sentiment analysis: Capturing favorability using natural language processing". In: *Proceedings of the 2nd international conference on Knowledge capture*, pp. 70–77.

Nguyen, Thien Hai and Kiyoaki Shirai (Sept. 2015). "PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 2509–2514. URL: https://www.aclweb.org/anthology/D15-1298.

Nivre, Joakim et al. (June 2007). "The CoNLL 2007 Shared Task on Dependency Parsing". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, pp. 915–932. URL: https://www.aclweb.org/anthology/D07-1096.

Øvrelid, Lilja et al. (May 2020). "A Fine-grained Sentiment Dataset for Norwegian". In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 5025–5033. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.618.

Paltoglou, Georgios and Mike Thelwall (July 2010). "A Study of Information Retrieval Weighting Schemes for Sentiment Analysis". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 1386–1395. URL: https://www.aclweb.org/anthology/P10-1141.

Pang, Bo and Lillian Lee (July 2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, pp. 271–278. URL: https://www.aclweb.org/anthology/P04-1035.

— (June 2005). "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 115–124. URL: https://www.aclweb.org/anthology/P05-1015.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (July 2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques". In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, pp. 79–86. URL: https://www.aclweb.org/anthology/W02-1011.

Passonneau, Rebecca J. (May 2004). "Computing Reliability for Coreference Annotation". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/752.pdf.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.

neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pedersen, Ted (2008). "Last Words: Empiricism Is Not a Matter of Faith". In: *Computational Linguistics* 34.3, pp. 465–470. URL: https://www.aclweb.org/anthology/J08-3010.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct, pp. 2825–2830. URL: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL: https://www.aclweb.org/anthology/D14-1162.

Pereg, Oren et al. (Nov. 2019). "ABSApp: A Portable Weakly-Supervised Aspect-Based Sentiment Extraction System". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, pp. 1–6. URL: https://www.aclweb.org/anthology/D19-3001.

Peters, Matthew et al. (July 2017). "Semi-supervised sequence tagging with bidirectional language models". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1756–1765. URL: https://www.aclweb.org/anthology/P17-1161.

Peters, Matthew et al. (June 2018a). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. URL: https://www.aclweb.org/anthology/N18-1202.

Peters, Matthew et al. (2018b). "Dissecting Contextual Word Embeddings: Architecture and Representation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1499–1509. URL: https://www.aclweb.org/anthology/D18-1179.

Pontiki, Maria et al. (Aug. 2014). "SemEval-2014 Task 4: Aspect Based Sentiment Analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 27–35. URL: https://www.aclweb.org/anthology/S14-2004.

Pontiki, Maria et al. (June 2015). "SemEval-2015 Task 12: Aspect Based Sentiment Analysis". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, pp. 486–495. URL: https://www.aclweb.org/anthology/S15-2082.

Pontiki, Maria et al. (June 2016). "SemEval-2016 Task 5: Aspect Based Sentiment Analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 19–30. URL: https://www.aclweb.org/anthology/S16-1002.

Popescu, Ana-Maria and Oren Etzioni (Oct. 2005). "Extracting Product Features and Opinions from Reviews". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 339–346. URL: https://www.aclweb.org/anthology/H05-1043.

Poria, Soujanya et al. (2016). "Fusing audio, visual and textual clues for sentiment analysis from multimodal content". In: *Neurocomputing* 174, pp. 50 –59. ISSN: 0925-2312. URL: http://www.sciencedirect.com/science/article/pii/S0925231215011297.

Poria, Soujanya et al. (2020). "Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research". In: *arXiv preprint arXiv:2005.00357*. URL: https://arxiv.org/pdf/2005.00357.pdf.

Potthast, Martin et al. (2016). "Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval". In: *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*. Ed. by Nicola Ferro et al. Vol. 9626. Lecture Notes in Computer Science. Springer, pp. 393–407. URL: https://doi.org/10.1007/978-3-319-30671-1_29.

Prinz, Florian, Thomas Schlange, and Khusru Asadullah (2011). "Believe it or not: how much can we rely on published data on potential drug targets?" In: *Nature reviews Drug discovery* 10.9, pp. 712–712.

Qian, Ning (1999). "On the momentum term in gradient descent learning algorithms". In: *Neural Networks* 12.1, pp. 145 –151. ISSN: 0893-6080. URL: http://www.sciencedirect.com/science/article/pii/S0893608098001166.

Qiu, Guang et al. (2011). "Opinion Word Expansion and Target Extraction through Double Propagation". In: *Computational Linguistics* 37.1, pp. 9–27. URL: https://www.aclweb.org/anthology/J11-1002.

Raaijmakers, Stephan, Khiet Truong, and Theresa Wilson (Oct. 2008). "Multimodal Subjectivity Analysis of Multiparty Conversation". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 466–474. URL: https://www.aclweb.org/anthology/D08-1049.

Radford, Alec et al. (2018). *Improving Language Understanding by Generative Pre-Training*. Tech. rep. OpenAI. URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, Alec et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI Blog* 1.8. URL: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Reimers, Nils and Iryna Gurevych (Sept. 2017). "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 338–348. URL: https://www.aclweb.org/anthology/D17-1035.

— (2018). "Why comparing single performance scores does not allow to draw conclusions about machine learning approaches". In: *arXiv preprint arXiv:1803.09578*. URL: https://arxiv.org/pdf/1803.09578.pdf.

Riedel, Benjamin et al. (2017). "A simple but tough-to-beat baseline for the Fake News Challenge stance detection task". In: *arXiv preprint arXiv:1707.03264*. URL: https://arxiv.org/pdf/1707.03264.pdf.

Rietzler, Alexander et al. (May 2020). "Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification". English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4933–4941. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.607.

Robertson, Stephen E et al. (1995). "Okapi at TREC-3". In: *Nist Special Publication Sp* 109, pp. 109,126. URL: https://bit.ly/33k98G6.

Rosenthal, Sara, Noura Farra, and Preslav Nakov (Aug. 2017). "SemEval-2017 Task 4: Sentiment Analysis in Twitter". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 502–518. URL: https://www.aclweb.org/anthology/S17-2088.

Rosenthal, Sara et al. (June 2015). "SemEval-2015 Task 10: Sentiment Analysis in Twitter". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, pp. 451–463. URL: https://www.aclweb.org/anthology/S15-2078.

Ruder, Sebastian (2019). "Neural transfer learning for natural language processing". PhD thesis. NUI Galway.

Ruder, Sebastian, Parsa Ghaffari, and John G. Breslin (Nov. 2016a). "A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 999–1005. URL: https://www.aclweb.org/anthology/D16-1103.

— (June 2016b). "INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 330–336. URL: https://www.aclweb.org/anthology/S16-1053.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/a164453.pdf.

Russo, Irene, Tommaso Caselli, and Carlo Strapparava (June 2015). "SemEval-2015 Task 9: CLIPEval Implicit Polarity of Events". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, pp. 443–450. URL: https://www.aclweb.org/anthology/S15-2077.

Saias, José (June 2015). "Sentiue: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 12". In: *Proceedings of the 9th International Workshop on Se-*

mantic Evaluation (SemEval 2015). Denver, Colorado: Association for Computational Linguistics, pp. 767–771. URL: https://www.aclweb.org/anthology/S15-2130.

Sanh, Victor, Thomas Wolf, and Sebastian Ruder (2019). "A hierarchical multi-task approach for learning embeddings from semantic tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 6949–6956.

Schwartz, Roy et al. (2019). "Green ai". In: *arXiv preprint arXiv:1907.10597*. URL: https://arxiv.org/pdf/1907.10597.pdf.

Snyder, Benjamin and Regina Barzilay (Apr. 2007). "Multiple Aspect Ranking Using the Good Grief Algorithm". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, pp. 300–307. URL: https://www.aclweb.org/anthology/N07-1038.

Socher, Richard et al. (July 2012). "Semantic Compositionality through Recursive Matrix-Vector Spaces". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pp. 1201–1211. URL: https://www.aclweb.org/anthology/D12-1110.

Socher, Richard et al. (Oct. 2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642. URL: https://www.aclweb.org/anthology/D13-1170.

Søgaard, Anders and Yoav Goldberg (Aug. 2016). "Deep multi-task learning with low level tasks supervised at lower layers". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 231–235. URL: https://www.aclweb.org/anthology/P16-2038.

Søgaard, Anders et al. (June 2014). "What's in a p-value in NLP?" In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 1–10. URL: https://www.aclweb.org/anthology/W14-1601.

Somasundaran, Swapna (2010). "Discourse-level relations for Opinion Analysis". PhD thesis. University of Pittsburgh. URL: http://d-scholarship.pitt.edu/8493/.

Somasundaran, Swapna, Josef Ruppenhofer, and Janyce Wiebe (June 2008a). "Discourse Level Opinion Relations: An Annotation Study". In: *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. Columbus, Ohio: Association for Computational Linguistics, pp. 129–137. URL: https://www.aclweb.org/anthology/W08-0122.

Somasundaran, Swapna and Janyce Wiebe (Aug. 2009a). "Recognizing Stances in Online Debates". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, pp. 226–234. URL: https://www.aclweb.org/anthology/P09-1026.

— (June 2010). "Recognizing Stances in Ideological On-Line Debates". In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Los Angeles, CA: Association for Computational Linguistics, pp. 116–124. URL: https://www.aclweb.org/anthology/W10-0214.

Somasundaran, Swapna, Janyce Wiebe, and Josef Ruppenhofer (Aug. 2008b). "Discourse Level Opinion Interpretation". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 801–808. URL: https://www.aclweb.org/anthology/C08-1101.

Somasundaran, Swapna et al. (Aug. 2009b). "Opinion Graphs for Polarity and Discourse Classification". In: *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*. Suntec, Singapore: Association for Computational Linguistics, pp. 66–74. URL: https://www.aclweb.org/anthology/W09-3210.

Somasundaran, Swapna et al. (Aug. 2009c). "Supervised and Unsupervised Methods in Employing Discourse Relations for Improving Opinion Polarity Classification". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp. 170–179. URL: https://www.aclweb.org/anthology/D09-1018.

Song, Youwei et al. (2019). "Attentional encoder network for targeted sentiment classification". In: *arXiv preprint arXiv:1902.09314*. URL: https://arxiv.org/pdf/1902.09314.pdf.

Stoyanov, Veselin and Claire Cardie (Aug. 2008). "Topic Identification for Fine-Grained Opinion Analysis". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 817–824. URL: https://www.aclweb.org/anthology/C08-1103.

Strubell, Emma et al. (Sept. 2017). "Fast and Accurate Entity Recognition with Iterated Dilated Convolutions". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2670–2680. URL: https://www.aclweb.org/anthology/D17-1283.

Sukthanker, Rhea et al. (2020). "Anaphora and coreference resolution: A review". In: *Information Fusion* 59, pp. 139 –162. ISSN: 1566-2535. URL: http://www.sciencedirect.com/science/article/pii/S1566253519303677.

Sun, Chi, Luyao Huang, and Xipeng Qiu (June 2019a). "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 380–385.

Sun, Chi et al. (2019b). "How to fine-tune BERT for text classification?" In: *China National Conference on Chinese Computational Linguistics*. Springer, pp. 194–206. URL: https://arxiv.org/pdf/1905.05583.pdf.

Sun, Kai et al. (Nov. 2019c). "Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5679–5688. URL: https://www.aclweb.org/anthology/D19-1569.

Sygkounas, Efstratios, Giuseppe Rizzo, and Raphaël Troncy (2016). "A Replication Study of the Top Performing Systems in SemEval Twitter Sentiment Analysis". In: *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*. Ed. by Paul T. Groth

et al. Vol. 9982. Lecture Notes in Computer Science, pp. 204–219. URL: https://doi.org/10.1007/978-3-319-46547-0_22.

Szegedy, Christian et al. (2016). "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, pp. 2818–2826. URL: https://doi.org/10.1109/CVPR.2016.308.

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning (July 2015). "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1556–1566. URL: https://www.aclweb.org/anthology/P15-1150.

Tang, Duyu, Bing Qin, and Ting Liu (July 2015). "Learning Semantic Representations of Users and Products for Document Level Sentiment Classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1014–1023. URL: https://www.aclweb.org/anthology/P15-1098.

— (Nov. 2016a). "Aspect Level Sentiment Classification with Deep Memory Network". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 214–224.

Tang, Duyu et al. (June 2014). "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 1555–1565. URL: https://www.aclweb.org/anthology/P14-1146.

Tang, Duyu et al. (Dec. 2016b). "Effective LSTMs for Target-Dependent Sentiment Classification". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 3298–3307. URL: https://www.aclweb.org/anthology/C16-1311.

Tang, Jialong et al. (July 2019). "Progressive Self-Supervised Attention Learning for Aspect-Level Sentiment Analysis". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 557–566.

Tay, Yi, Luu Anh Tuan, and Siu Cheung Hui (2018). "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPDFInterstitial/16570/16162.

Taylor, Ann, Mitchell Marcus, and Beatrice Santorini (2003). "The Penn treebank: an overview". In: *Treebanks*. Springer, pp. 5–22.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (July 2019). "BERT Rediscovers the Classical NLP Pipeline". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601. URL: https://www.aclweb.org/anthology/P19-1452.

Tjong Kim Sang, Erik F. and Sabine Buchholz (2000). "Introduction to the CoNLL-2000 Shared Task Chunking". In: *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*. URL: https://www.aclweb.org/anthology/W00-0726.

Toh, Zhiqiang and Wenting Wang (Aug. 2014). "DLIREC: Aspect Term Extraction and Term Polarity Classification System". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 235–240. URL: https://www.aclweb.org/anthology/S14-2038.

Toprak, Cigdem, Niklas Jakob, and Iryna Gurevych (July 2010). "Sentence and Expression Level Annotation of Opinions in User-Generated Discourse". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 575–584. URL: https://www.aclweb.org/anthology/P10-1059.

Tsai, Henry et al. (Nov. 2019). "Small and Practical BERT Models for Sequence Labeling". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3632–3636. URL: https://www.aclweb.org/anthology/D19-1374.

Turian, Joseph, Lev-Arie Ratinov, and Yoshua Bengio (July 2010). "Word Representations: A Simple and General Method for Semi-Supervised Learning". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 384–394. URL: https://www.aclweb.org/anthology/P10-1040.

Turney, Peter (July 2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 417–424. URL: https://www.aclweb.org/anthology/P02-1053.

Varathan, Kasturi Dewi, Anastasia Giachanou, and Fabio Crestani (2017). "Comparative opinion mining: A review". In: *Journal of the Association for Information Science and Technology* 68.4, pp. 811–829. eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23716. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.23716.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008. URL: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Vecchio, Marco Del et al. (2018). "The Data Science of Hollywood: Using Emotional Arcs of Movies to Drive Business Model Innovation in Entertainment Industries". In: *ArXiv* abs/1807.02221. URL: https://arxiv.org/pdf/1807.02221.pdf.

Viani, Natalia et al. (Oct. 2018). "Time Expressions in Mental Health Records for Symptom Onset Extraction". In: *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*. Brussels, Belgium: Association for Computational Linguistics, pp. 183–192. URL: https://www.aclweb.org/anthology/W18-5621.

Vo, Duy-Tin and Yue Zhang (2015). "Target-dependent twitter sentiment classification with rich automatic features". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. URL: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/viewFile/10707/10850.

Wang, Bo et al. (Apr. 2017a). "TDParse: Multi-target-specific sentiment recognition on Twitter". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 483–493. URL: https://www.aclweb.org/anthology/E17-1046.

Wang, Bo et al. (Nov. 2017b). "TOTEMSS: Topic-based, Temporal Sentiment Summarisation for Twitter". In: *Proceedings of the IJCNLP 2017, System Demonstrations*. Tapei, Taiwan: Association for Computational Linguistics, pp. 21–24. URL: https://www.aclweb.org/anthology/I17-3006.

Wang, Hongning, Yue Lu, and ChengXiang Zhai (2010). "Latent aspect rating analysis on review text data: a rating regression approach". In: *KDD '10*.

Wang, Jingjing et al. (2018). "Aspect Sentiment Classification with both Word-level and Clause-level Attention Networks." In: *IJCAI*. Vol. 2018, pp. 4439–4445.

Wang, Jingjing et al. (Nov. 2019). "Human-Like Decision Making: Document-level Aspect Sentiment Classification via Hierarchical Reinforcement Learning". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5581–5590. URL: https://www.aclweb.org/anthology/D19-1560.

Wang, Sida and Christopher Manning (July 2012). "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, pp. 90–94. URL: https://www.aclweb.org/anthology/P12-2018.

Wang, Wenya et al. (Nov. 2016a). "Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 616–626. URL: https://www.aclweb.org/anthology/D16-1059.

— (2017c). "Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder P. Singh and Shaul Markovitch. AAAI Press, pp. 3316–3322. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14441.

Wang, Yequan et al. (Nov. 2016b). "Attention-based LSTM for Aspect-level Sentiment Classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 606–615. URL: https://www.aclweb.org/anthology/D16-1058.

Wang, Zihan, Stephen Mayhew, Dan Roth, et al. (2020). "Cross-Lingual Ability of Multilingual BERT: An Empirical Study". In: *arXiv preprint arXiv:1912.07840*. URL: https://arxiv.org/pdf/1912.07840.pdf.

Webber, Bonnie et al. (2003). "Anaphora and Discourse Structure". In: *Computational Linguistics* 29.4, pp. 545–587. URL: https://www.aclweb.org/anthology/J03-4002.

Welch, B. L. (1947). "The Generalization of 'Student's' Problem when Several Different Population Variances are Involved". In: *Biometrika* 34.1/2, pp. 28–35. ISSN: 00063444. URL: http://www.jstor.org/stable/2332510.

Weston, Jason (2007). "Large-Scale Semi-Supervised Learning". In: *NATO ASI Mining Massive Data Sets for Security*. URL: https://bit.ly/2WTB4zt.

Weston, Jason, Sumit Chopra, and Antoine Bordes (2015). "Memory Networks". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1410.3916.

Whitelaw, Casey, Navendu Garg, and Shlomo Argamon (2005). "Using appraisal groups for sentiment analysis". In: *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 625–631.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie (2005). "Annotating expressions of opinions and emotions in language". In: *Language resources and evaluation* 39.2-3, pp. 165–210.

Wiebe, Janyce M. (1994). "Tracking Point of View in Narrative". In: *Computational Linguistics* 20.2, pp. 233–287. URL: https://www.aclweb.org/anthology/J94-2004.

Wiegand, Michael and Dietrich Klakow (Apr. 2012). "Generalization Methods for In-Domain and Cross-Domain Opinion Holder Extraction". In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association for Computational Linguistics, pp. 325–335. URL: https://www.aclweb.org/anthology/E12-1033.

Wilcoxon, Frank (1945). "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6, pp. 80–83. ISSN: 00994987. URL: http://www.jstor.org/stable/3001968.

Williams, Ronald J (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3-4, pp. 229–256.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann (Oct. 2005). "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 347–354. URL: https://www.aclweb.org/anthology/H05-1044.

Wilson, Theresa Ann (2008). "Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states". PhD thesis. University of Pittsburgh. URL: http://d-scholarship.pitt.edu/id/eprint/7563.

Xiang, Chunli, Yafeng Ren, and Donghong Ji (2019). "Identifying Implicit Polarity of Events by Using an Attention-Based Neural Network Model". In: *IEEE Access* 7, pp. 133170–133177.

Xu, Hu et al. (July 2018). "Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia:

Association for Computational Linguistics, pp. 592–598. URL: https://www.aclweb.org/anthology/P18-2094.

Xu, Hu et al. (June 2019). "BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2324–2335.

Xu, Jiacheng et al. (Nov. 2016). "Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1660–1669. URL: https://www.aclweb.org/anthology/D16-1172.

Xue, Wei and Tao Li (July 2018). "Aspect Based Sentiment Analysis with Gated Convolutional Networks". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2514–2523. URL: https://www.aclweb.org/anthology/P18-1234.

Yang, Bishan and Claire Cardie (Aug. 2013). "Joint Inference for Fine-grained Opinion Extraction". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1640–1649. URL: https://www.aclweb.org/anthology/P13-1161.

— (June 2014). "Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 325–335. URL: https://www.aclweb.org/anthology/P14-1031.

Yang, Jun et al. (2018). "Multi-entity aspect-based sentiment analysis with context, entity and aspect memory". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17036/16171.

Yang, Zhilin et al. (2019). "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems*, pp. 5754–5764. URL: https://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf.

Yang, Zichao et al. (June 2016). "Hierarchical Attention Networks for Document Classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 1480–1489. URL: https://www.aclweb.org/anthology/N16-1174.

Yannakoudakis, Helen et al. (Sept. 2017). "Neural Sequence-Labelling Models for Grammatical Error Correction". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2795–2806. URL: https://www.aclweb.org/anthology/D17-1297.

Yin, Yichun, Yangqiu Song, and Ming Zhang (Sept. 2017). "Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2044–2054. URL: https://www.aclweb.org/anthology/D17-1217.

Zeng, Biqing et al. (2019). "LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification". In: *Applied Sciences* 9.16, p. 3389. URL: https://www.mdpi.com/2076-3417/9/16/3389/htm.

Zhang, Chen, Qiuchi Li, and Dawei Song (Nov. 2019a). "Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 4568–4578. URL: https://www.aclweb.org/anthology/D19-1464.

Zhang, Lei and Bing Liu (June 2011). "Identifying Noun Product Features that Imply Opinions". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 575–580. URL: https://www.aclweb.org/anthology/P11-2101.

Zhang, Meishan, Peili Liang, and Guohong Fu (June 2019b). "Enhancing Opinion Role Labeling with Semantic-Aware Word Representations from Semantic Role Labeling". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 641–646. URL: https://www.aclweb.org/anthology/N19-1066.

Zhang, Meishan, Yue Zhang, and Duy-Tin Vo (Sept. 2015a). "Neural Networks for Open Domain Targeted Sentiment". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 612–621. URL: https://www.aclweb.org/anthology/D15-1073.

— (2016). "Gated neural networks for targeted sentiment analysis". In: *Thirtieth AAAI Conference on Artificial Intelligence*. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/viewFile/12074/12065.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015b). "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems*, pp. 649–657. URL: https://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf.

Zhang, Yuan and Yue Zhang (July 2019c). "Tree Communication Models for Sentiment Analysis". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3518–3527. URL: https://www.aclweb.org/anthology/P19-1342.

Zhao, Pinlong, Linlin Hou, and Ou Wu (2019). "Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification". In: *Knowledge-Based Systems*, p. 105443.

Zhu, Xiaojin Jerry (2005). *Semi-supervised learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences.

Zhuang, Li, Feng Jing, and Xiao-Yan Zhu (2006). "Movie Review Mining and Summarization". In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. CIKM '06. Arlington, Virginia, USA: Association for Computing Machinery, 43–50. ISBN: 1595934332. URL: https://doi.org/10.1145/1183614.1183625.