Evaluating Sentiment Analysis Approaches Across Traditional and Novel Datasets

Oliver Hall
MSci Hons. Computer Science

May 2021

Declaration

I certify that the material contained within this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted version of this submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work. I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Date: 18/05/2021
Signed: Oliver Hall

Project Supervisor: Professor Paul Rayson
Working documents available at:

Abstract

This dissertation looks to evaluate various cutting-edge Natural Language Processing (NLP) approaches to explore their strengths, pitfalls and potential future applications within the realm of Sentiment Analysis. Though research of Sentiment Analysis has seen a massive increase in the past five years, including a heightened focus on Aspect-Based analysis, we seemingly remain no closer to finding a clear "best" methodology for extracting deeper meaning from text. Exploration of fictional corpora through script excerpts, as well as traditional datasets such as online reviews, pushes these tools to their limit, with the main focus on evaluating models for advanced sentiment analysis when utilising text that differs from the standard "review" format.

Contents

List of Figures

List of Tables

1 Introduction

The aim of this dissertation is to comparatively evaluate the different types of widely used Sentiment Analysis methods currently available, which range from simple rule-based systems to advanced deep learning models.

1.1 Project Motivation

Driven primarily by an increasing dependency on online textual resources, Natural Language Processing (NLP) has continued to grow as a discipline in recent years as computational approaches are pushed to the forefront of modern text analysis. Inspired primarily by the constant migration of customer interactions onto digital formats via comments, reviews and surveys, Sentiment Analysis (SA) has become an integral part of data analytics for companies within growing e-commerce markets. There is a large demand for systems capable of reading, summarising and extracting key information from large amounts of data, so that companies can map the opinions of their customers and adapt products and services accordingly. However, although it has become an increasingly prevalent and integral part of market analytics for organisations in recent years, SA is far from a novel concept. Not only have the recent advances towards automated SA systems been created through improved computing capabilities to handle big data and the availability of quick customer interaction, but the very purpose of SA has shifted from analysing online product reviews to social media texts (Mäntylä et al. 2018). Many topics beyond product reviews, such as cyberbullying (Naf'an et al. 2019) and Fake News detection (Bhutani et al. 2019), now extend the utilisation of SA (Piryani et al. 2017). It is this new domain of SA, beyond examination of product reviews, that is of particular interest within this project.

The methods for conducting large-scale SA continue to develop and evolve, with over 7,000 academic articles published on the topic in the last 15 years (Mäntylä et al. 2018). However, although individual evaluations exist for many of the cutting-edge tools in development, there is a distinct lack of comparative evaluations between these methods. There also exists a distinct gap in the types of data being examined with SA, which is usually tested on reviews of restaurants and films. For example, there is a lack of academic work surrounding perceptions of characters in written text, where the perceived value of 'sentiment' may be significantly more implicit than a much more obvious statement of "the food was great" within a restaurant review. The possible benefits of creating systems that can actively track deeper, implicit sentiment such as how a character may be perceived by an audience before a book or film is even published soon become apparent; a story-teller can alter their work accordingly before it is ever released into the public domain. This new way of examining sentiment goes beyond the current use cases of commercial opinion mining, and looks to form a basis for how computing systems interact with and understand text.

Further to the point, the need to evaluate current SA tools on a fictional work was clear from the offset. The TV show *Friends* was chosen as a basis for conducting sentiment analysis in the fictional domain, as it is a self-contained series with no spin-off books or other related material. This allows us to conclusively state that for any sentiments expressed about the show, be that in the form of Tweets or IMDb reviews, the opinions are more likely to be expressed only in reference to the show material. The use of the scripts from the show also presents a very different type of domain, differing drastically from the typical review format. In light of this, in addition to a traditional movie review dataset, it was deemed interesting to evaluate SA tools against a dataset of IMDb reviews from the TV show *Friends*, excerpts of scripts from the show and online Twitter reviews of the show. This research would hopefully allow us to examine the complications when performing cross-domain sentiment analysis.

In light of this, and having conducted a Third Year Project examining how NLP tools function when analysing a fictional text, the goal of the project is to evaluate the strengths and weaknesses of currently available SA methods when used in a traditional sense on written reviews, as well as in the more novel context of tracking character perceptions directly from a TV script. There was also an initial interest, fuelled by work carried out in the SCC.419 Industrial Placement module to develop a sophisticated Twitter-scraping technology, in examining how sentiment can be expressed in different ways for different reviewing mediums. The question was posed as to whether users on Twitter would express their views in more extreme forms than in a formal review setting, and more importantly, whether current SA systems would be able to notice this difference in subtlety.

1.2 Aims

The overall goals of the project are outlined as follows:

1. Provide a detailed overview of current Sentiment Analysis methods, including strengths and shortcomings, as well as how these approaches have developed over time.

2. Identify and explain individual tools for each type of method which can then be implemented for testing.

3. Evaluate each of these tools against a large IMDb[1] movie reviews dataset, individual IMDb reviews and the full anthology of scripts from the TV show *Friends*.

4. Explore how the tools perceive opinion when comparing human perception graphs of sentiment with opinions expressed on Twitter, to gain an understanding of how they evaluate subtlety.

5. Explore the reasoning behind the conclusions drawn through testing before making inferences on the best use cases for each method and how the tools could be improved in future.

6. Make references to how this information can be best utilised within the wider field of NLP.

1.3 Report Structure

This report will explore background research into Natural Language Processing and its subbranch of Sentiment Analysis. It will examine past publications relating to the field of SA, including a detailed overview of how methods have evolved in recent years. The key concepts and principles of SA will be outlined in full, so that a base understanding can be maintained throughout the report. The current roadblocks within the field of SA will also be explored before various methods and approaches to the problem are outlined. There will be a justification of why each tool was chosen to represent a certain approach followed by an explanation of other lines of inquiry that were followed but ultimately not included within the evaluation stage. Beyond this, the datasets will be explained, including a detailed pipeline of how they were collated and pre-processing steps carried out on them. The performance of each tool for each dataset will then be presented and explained, which will allow for a deep comparative evaluation within the Results chapter. A purposeful conclusion will be drawn to the project, including a return to the aims outlined in Chapter 1.2, references to how the work affects the wider scientific community and a discussion of any potential further avenues of research.

[1]Internet Movie Database available at https://www.imdb.com/

2 Background

This chapter will describe Natural Language Processing as a discipline before delving into Sentiment Analysis in the context of the project. A brief timeline of the field will be outlined, including how the approaches have changed over time and the current issues. There will also be a brief explanation of each tool evaluated within the project, including why they were chosen over other options.

2.1 Natural Language Processing

Though often considered a novel area of artificial intelligence, NLP has been a topic of research since just after the Second World War under the initial idea of converting one human language to another (dubbed Machine Translation Delavenay & Delavenay (1960)). Between the 1980s and 1990s the study area of computational grammar continued to grow as an active field of research and was joined by other essential topics such as automatic summarisation (Hahn & Mani 2000), statistical language processing (Dror et al. 2018) and information extraction (Ciravegna et al. 2001). In current terms, NLP encapsulates a range of functions from parsing (Klein & Manning 2003) and part-of-speech tagging (Voutilainen 2003) to advanced machine learning (Le Glaz et al. 2021) and deep learning technologies (Landolt et al. 2021).

Examples of modern applications for NLP include health analytics (Vaira et al. 2018), where models can predict the presence of over 100 different diseases using pattern recognition methods and patient speech, as well as conversational framework applications (Mnasri 2019) in the form of popular smart devices, such as Amazon's Alexa[2] and Apple's Siri[3]. Another key application and perhaps the biggest commercial use of NLP is in the form of Sentiment Analysis.

2.2 Sentiment Analysis

As a subfield of NLP, SA (sometimes referred to as Opinion Mining) aims to extract opinions, attitudes and emotions from text. The overall objective of SA is generally considered to be a classification problem, whereby text should be objectively ranked on the positive to negative spectrum. Traditionally, SA was carried out by a large labour force who would read through and manually assess text - a costly approach prone to human error. Though this approach is still utilised in some cases, the rise of modern computing power has made the practice much rarer. Although automated computing methods remove the need for hundreds of workers and hours upon hours of time, they face an inherent problem in that people express and interpret sentiment intensity and polarity differently. Known as polysemy, words can have many possible meanings based on the context of a sentence, which creates a need for computing models with a deeper understanding of language in order to obtain more sophisticated results (Marcus 2020).

Common SA tools operate on two key methods: lexicon rule-based and automated learning approaches. Lexicon rule-based methods use pre-defined lists of words and their associated polarity to quickly provide sentiment values for every word within a text, before finding an average.

[2]Alexa technology further outlined at https://developer.amazon.com/en-GB/alexa

[3]Siri technology available at https://www.apple.com/uk/siri/

	pos	neg	neu	total
today			0 + *1*	normalizing function: $\frac{x}{\sqrt{x^2+\alpha}}$
is			0 + *1*	
a			0 + *1*	
good	1.9 + *1*			
day			0 + *1*	
	2.9		4	6.9
	2.9 / 6.9	0 / 6.9	4 / 6.9	1.9 / ((1.9^2) + *15*)^0.5
	0.42	0	0.58	0.44
+1 to compensate for neutral words				
15 is the approximate max sentiment score				

Figure 1: Example of rule-based sentiment analysis (Lee. 2021)

Figure 1 shows this practice in action, with each word in the phrase being assigned a polarity score from a predefined lexicon. From there, an average sentiment can be found, including a normalising function to account for neutral values. The approximate maximum sentiment score of 15 (assuming each word scores a positive of 2, with an added 1 to account for neutral words) is used within the normalising function to find a weighted average. x refers to the total positive weightings of each individual word within the sentence, minus the neutral compensation values, and a refers to the approximate maximum sentiment score. For this example, the final weighted average of the sentence would be 0.44.

However, although these methods were some of the first SA systems to be developed, rule-based models struggle when it comes to the pragmatics and polysemy of words. For example, phrases such as "You're on fire!" may be rated negatively when in reality the meaning behind the statement is positive (Asghar et al. 2017). It is this level of naivety and lack of understanding of deeper context within a sentence, such as sarcasm and irony (Maynard & Greenwood 2014), that has pushed academics towards automated learning models.

Recent automated approaches use training and testing to build models for more in-depth SA (Hasan et al. (2018), Ghiassi & Lee (2018), Hew et al. (2020)). Through the processes of training, prediction, evaluation and re-training, these models can go through many iterations to build up and "learn" how to evaluate the sentiment within text. Within the training process, the model *learns* to associate inputs with a corresponding output, which can then be evaluated against a set of unseen testing data. The models can evaluate their performance to identify where they may have mis-classified the sentiment of words or phrases, and feed this into the next iteration of training. These models are adaptable, in that they can be trained on data from different domains so that they can be built to identify sentiment within a very specific context, i.e. movie or restaurant reviews. This adaptability, accompanied by the range of customisation available when fine-tuning a learning model, is directly attributed to the rise in popularity of automated processing within both SA and NLP as a whole.

As well as having different approaches for deriving sentiment, SA also features varying levels of granularity on which to analyse text. Approaches operate primarily on three different levels - document, sentence and aspect levels - outlined below.

2.2.1 Document-Level SA

At document-level, SA is used to determine the overall opinion of a text. Before finer granularity systems were developed, document-level analysis was the favoured approach for long-form sentiment analysis when evaluating reviews and customer feedback (Choi et al. 2020). There is, however, a key assumption being made in order to achieve document-level analysis - that the

entirety of each document only expresses opinions on a single entity. This makes document-level SA incredibly difficult when evaluating a large dataset containing multiple possible entities for opinions to be expressed about, such as a movie reviews dataset where reviewers may comment on the actors, writing, cinematography, etc. Furthermore, even relatively advanced machine learning methods for document-level SA do not achieve good performance when attempting sentiment classification (Liu & Zhang (2012), Behdenna et al. (2018)).

Taking these drawbacks into consideration, document-level analysis is considered in modern terms to be too reductive, as it aims to summarise an entire block of text into a single polarity value, likely missing the deeper levels of nuance present within the material. As such, document-level approaches will not be examined throughout this report.

2.2.2 Sentence-Level SA

Rather than evaluating an entire document and producing a single output (to classify a customer review as positive or negative for example), sentence-level SA methods were quickly developed in the early 2000s as a means of finding aggregate scores for individual sentences within a text (Meena & Prabhakar 2007). By utilising technologies such as NLTK's Sentence-Tokeniser[4], a document can easily be broken down into its component sentences. From there, with rule-based SA, each word in the sentence can be represented by its corresponding pre-defined value for sentiment (usually between -1 and 1) before the average score for the sentence is found. Although often incredibly useful for providing rough estimates of sentiment, sentence-level tools struggle when examining individual features of a review.

Figure 2: Example of sentence-level sentiment analysis

For example, if we were looking to examine the quality of food within a review from Figure 2, we would see that although the food was rated highly, the overall sentiment of the sentence may be returned as negative, due to the negative references to the staff. It is this shallow level of understanding that shows immediate flaws in the output of sentence-level tools. A full analysis of a sentence-level rule-based approach will be featured later in this report.

2.2.3 Aspect-Level SA

In order to combat the issues presented above, aspect-level systems have become increasingly prevalent in recent years. Aspect-level SA systems work at a finer level of granularity by breaking down a sentence into subclauses, where opinions can be directly attributed to entities within that subclause. Published as a key task in the SemEval 2016[5] competition (Pontiki et al. 2016), the concept (relatively novel at the time) was brought onto the radar of many NLP researchers, attracting 245 submissions for how best to implement an aspect-level system. It is worth noting at this point that there is a distinct difference between aspect-based SA (Pavlopoulos (2014), Thet et al. (2010)) and target-based SA (Li et al. (2019), Chen et al. (2018)); the former can refer to a category of elements, such as 'food', consisting of any references to apples, pears, etc, whereas target-based SA is concerned with a single entity. Aspect categories are chosen and defined for a specific domain, e.g., if an SA system was built to evaluate restaurant reviews, it

[4]Natural Language ToolKit tokeniser library available for further reading at https://www.nltk.org/_modules/nltk/tokenize.html

[5]SemEval (Semantic Evaluation) is an ongoing series of evaluations for semantic analysis systems. Workshops are held annually with a new, state of the art topic to evaluate every year

may be beneficial to define categories such as 'food', 'staff' or 'atmosphere', which may not be applicable to a film review domain. Within the context of this project, aspect-based SA will be used when evaluating the IMDb review dataset, as it provides a larger scope of responses. For example, a user might not explicitly say "the performance of Alice was good and the performance of Bob was good" but may instead refer to the 'actors' as 'great'. On the other hand, we will use a target-based approach when looking to evaluate responses to individual characters in later experiments, where it is important to be able to verify that an opinion is being expressed about a single entity, rather than any entity within the encapsulating category.

Figure 3: Example of aspect-level sentiment analysis

With reference to Figure 3, we can now see how an aspect-level SA approach may be able to separate the sentence into subclauses, and evaluate the descriptions for each aspect on an individual basis.

2.3 Existing Issues

One key issue within the current field of SA is the overwhelming complexity of not just classifying data, but having a system explain why it has classified data in a certain way. Previously, researchers have posited that the task of SA is actually broken down into subtasks, including **aspect extraction**, **aspect sentiment classification** and **opinion extraction** (Liu 2012). To better explain these, we will look at the example sentence "The staff are very friendly but the food is incredibly bad". The **aspect extraction** should identify "staff" and "food" as aspects, and the **aspect sentiment classification** should be positive and negative sentiment respectively. The **opinion extraction** should identify the words "friendly" and "bad", as these represent the opinionated comments that influence the sentiment. For a long time, these three subtasks were carried out individually (Kobayashi et al. 2004) (Chen et al. 2013). For example, a system may be able to identify the classification of sentiment incredibly well, but would struggle when trying to explain *why* by highlighting the opinions. Cutting-edge research carried out in the past 18 months has shown promising results when attempting to complete all three tasks as a whole, but still struggles to be consistent when handling complex opinions and grammar, as well as deeper subtleties such as sarcasm (Peng et al. 2020). In light of this, all tools evaluated within this report will only be evaluated by their accuracy in predicting positive and negative sentiment in reviews, rather than on their justifications of why they have classified the data in such a way.

Another primary drawback of increasingly complex supervised machine learning models for SA is the need for large, annotated datasets for training.. The fact that a model can be built for examining sentiment within a specific domain is incredibly advantageous as it highlights a step towards context-specific systems, but for each new realm of data being explored, there is a need for a large amount of pre-labelled training data. A recent article (Nazir et al. 2020) finds that the accuracy in SA systems is compromised when crossing domains and handling new, unlabelled data - e.g., a model trained on restaurant data may not be applicable to hotel reviews. This does however serve as an excellent justification for pushing current approaches to their limits, and influenced the decision to try and evaluate sentiment in nontraditional domains later in the project.

Not only are datasets domain specific, but currently there are only a handful[6] of key datasets suitable for use when training and evaluating SA models. The MPQA Opinion Corpus (Wiebe

[6]Full list of datasets available at https://paperswithcode.com/datasets?task=sentiment-analysis

et al. 2005) contains over 500 news articles, manually annotated to highlight opinions, beliefs, emotions and speculations. Though the dataset was an extraordinary contribution at the time of release, it is very likely that language has evolved in the previous 16 years and statements that were once considered "strong positive opinions" may not be labelled as such in the current day. The creators of the corpus also acknowledge the likelihood of bias in how the articles were annotated, stating that "in the longer term ... bias and point of view must be considered" (Wiebe et al. 2005). The other key dataset used for training (and one that features in the analysis of this report) is the IMDb Movie Reviews dataset (Maas et al. 2011). It attempts to combat potential bias by using a binary classifier to label reviews as positive (a rating of >6 out of 10) or negative (a rating of <5 out of 10). Scores of 5 and 6 are classified as neutral and not added to the dataset. Though this classification method has removed the need for a manual annotation of the data, it is also likely to have generated imperfections in the dataset - a review of 7 out of 10 could easily be considered a "neutral" review, given the average film score on the IMDb website is 6.7 out of 10, but in this case the review would be classified as "positive". This discrepancy could help to explain Figure 4, which shows how even state-of-the-art models are yet to achieve a 100% accuracy score. Each data point on the figure represents an attempt to generate an accurate SA approach, with the corresponding accuracy score on the y-axis. It is also worth noting that there has been no huge leaps in progress regarding this task in the past 2 years, which could indicate that available systems have reached the limits of advancement within this dataset. In the context of the project, evaluations of SA tools on the IMDb dataset will have adjusted expectations.

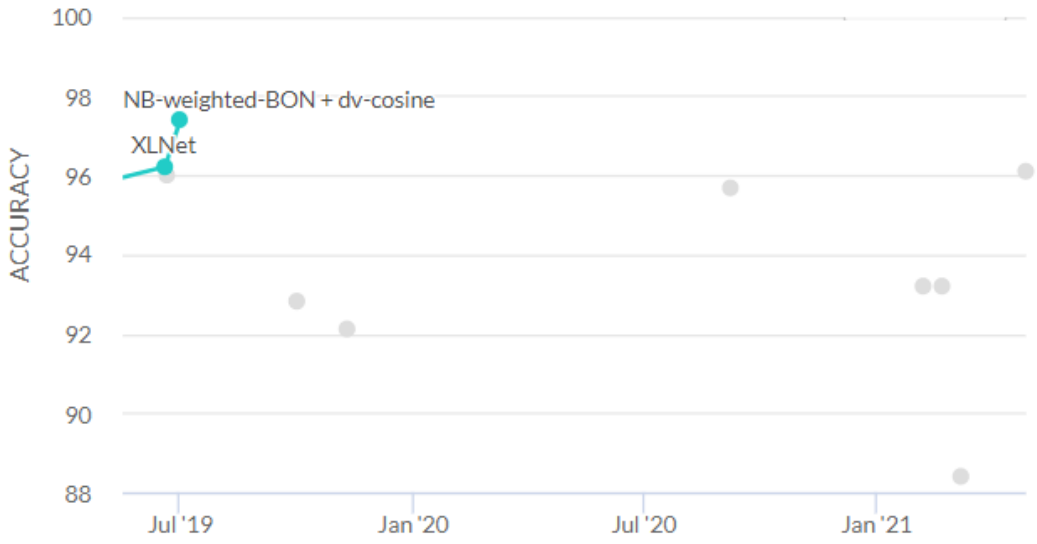

Figure 4: Current SOTA for Sentiment Analysis on IMDb dataset

2.4 Selection of Approaches

This section outlines each tool selected for evaluation within this report, including a detailed justification of why it was chosen and the type of approach that it takes to SA.

2.4.1 VADER: Rule-Based SA

An excellent example of a current lexicon-based tool is the Valence Aware Dictionary and sEntiment Reasoner (VADER, Hutto & Gilbert (2014)). Built and trained primarily on social media content and movie reviews, the tool was established a good addition to the project, given the IMDb review dataset that will be used in testing. Through previous experience

of using VADER to evaluate sentiment on a sentence-level (Hall 2020), it was apparent that the rule-based tool was significantly better than first predicted when compared with similar approaches such as TextBlob[7]. Suffering from a very minor speed-performance tradeoff, the tool has previously been rated as good as human-drawn perceptions when evaluating reviews (Hutto & Gilbert 2014), and is a great example of a widely used, highly regarded sentence-level SA tool.

The VADER lexicon contains a predetermined dictionary of 9000 tokens with sentiment scores between -4 and 4 for each feature (it is worth noting that it even supports emoticons and some acronyms for advanced SA in this field - e.g., the lexicon contains popular text based emoticons such as ':-)' with a corresponding sentiment score). It then uses these predefined scores to produce an overall compound score for the input text. It was decided that with a lexicon built specifically for online domains such as Twitter data, and as a tool that strongly outperforms other available rule-built implementations (Ribeiro et al. 2016), VADER would be an excellent addition to the analysis stage of the project. It has been noted that VADER may struggle when evaluating formal text, such as scientific papers or well-written film reviews due to its lexicon of online terminology (Elbagir & Yang 2019). The dictionary of 9000 tokens could also be seen as a limitation to the tool as if a given word is not present in the lexicon, it will be assigned a neutral score, when the word could simply be irregular and unaccounted for. Conversely, VADER is only 7 years old, and the lexicon is updated often[8], so the issue of expired terminology and language is not likely to significantly decrease performance within this project.

2.4.2 ABSA: Machine Learning SA

The second approach under scrutiny is the Aspect-Based Sentiment Analysis Tool, which utilises machine learning (ML) and aspect-level granularity. It is worth noting at this stage the difference between ML and deep learning approaches. Where ML uses algorithms to parse data, learn from the data and make an informed decision based on what it has parsed, deep learning is an advanced subfield of ML, which structures algorithms in layers to create an 'artificial neural network' that can make intelligent decisions on its own. Within the project, we will evaluate an implementation of both approaches - ABSA for ML and Keras for deep learning.

Though many standalone implementations of these aspect-level technologies exist (Moore & Rayson 2018), they tend to be incredibly domain specific. Though potentially seen as a weakness of the approach (an ML model trained on restaurant data may struggle to evaluate movie reviews), this domain specificity highlights a key strength of ML approaches - the ability to build custom models giving higher levels of control within a specific context. However, as supervised ML approaches require training data to be labelled, any subjectivity and bias present in the training data would likely also be reflected in the model (an example of this is outlined in Chapter 2.3). There also exists a great shortage of domain-specific training data. Due to this, the expected approach in recent years has shifted to one of models pre-trained on large scale language datasets, which are then fine-tuned for a specific domain.

Unlike the VADER model which has been trained for specific use on social media texts, the implementation of ABSA is built by default on an adaptation of the BERT model (Gao et al. 2019), fine-tuned for aspect-target sentiment classification (Rietzler et al. 2019). Trained on a restaurant review and laptop review dataset as presented in SemEval 2014 Task 4 (Pontiki et al. 2016), the model was built from manually annotated data with appropriate aspect-targets highlighted. The ABSA implementation utilises the BERT_BASE model; created on a bidirectional architecture, the BERT model can make more advanced inferences from a text

[7]TextBlob Text Processing available at https://textblob.readthedocs.io/en/dev/index.html

[8]Updates to VADER shown on the official GitHub page https://github.com/cjhutto/vaderSentiment

by exploring the relationships between all words within a sentence, rather than reading from left-to-right (Devlin et al. 2019). More information about the inner workings of the BERT model are available in (Rietzler et al. 2019).

With accuracy scores of above 80% when tested on the SemEval 2014 evaluation dataset (Pontiki et al. 2016), the ABSA implementation was identified as a key ML approach for evaluation within the project, as it bridges the gap between the rigid, rule-based approaches discussed previously and the more advanced, deep learning models. In (Rietzler et al. 2019), it is posited that the main shortcomings of the model include an emphasis on knowledge-based corpora (such as Wikipedia) for training of the BERT_BASE model, rather than corpora containing opinions - a key feature in sentiment analysis. Another weakness of the model is inconsistency in identifying which aspect-target an opinion belongs to, which could be counteracted by training on a larger amount of supervised data. These two shortcomings may explain where the remaining 20% of data is being incorrectly classified. It is hoped that the granularity of aspect-level present in ABSA would heighten the performance of the model when tested on our evaluation datasets, and add a lot of value to the discussion section of the report.

2.4.3 Keras: Deep Learning SA

The Recurrent Neural Network (RNN) model is a deep learning approach that operates on a sentence-level of granularity. Whereas rule-based solutions are rigid and limited by a lexicon that could expire, deep learning solutions associate input texts to corresponding output tags algorithmically. During training, pairs of texts and their manually labelled sentiment tags are fed into the deep learning algorithm. This leads to the creation of a model capable of making predictions on unseen texts. The unique feature of RNNs is their ability to use reasoning about previous events in a text to inform later ones. The term 'Keras' refers to the specific implementation of RNN utilised within the project.

RNNs contain loops within their pipeline, which allows information to persist between layers of repeated training. To simplify, RNNs can be thought of as multiple copies of the same network, each passing a message to a successor. Applications of RNNs include speech recognition (Yaxiong et al. 2014), translation (Hermanto et al. 2015), image captioning (Al-Muzaini et al. 2018) and language modelling (Soutner & Müller 2013).

Our Keras implementation of an RNN utilises Long-Short Term Memory (LSTM) technology (Hochreiter & Schmidhuber 1997), which are capable of learning long-term dependencies. For example, if we try to predict the last word in the text "I was born in Spain. I speak fluent *Spanish*", we may be able to predict that the last word should be the name of a language. However, for a system to predict which language, it would need the context of the first sentence. LSTMs are designed to avoid these long-term dependency problems.

LSTMs contain a central "cell state", which houses information that will be passed into the next iteration of the RNN. This cell state can be altered by three optional flows of information called gates (composed of a sigmoid neural net layer and a pointwise multiplication operation). The gates output numbers between zero and one, referring to how much information should be allowed through. Gate one, or the 'forget layer', refers to how much information from the previous iteration should be thrown away. Gate two, or 'input layer', decides which values will be updated from the previous iteration, and Gate three, or 'candidate layer', provides predictions of new possible values to replace those chosen by Gate two. Further information on how these layers work is available in the work of Nowak et al. (2017).

When originally tested in (Yu et al. 2019) on the IMDb dataset, the Keras LSTM implementation produced an accuracy score of 89%. We will look to reproduce this score, before evaluating the tool on cross-domain datasets. Within the context of this project, the important differentiation between the Keras implementation and other tools is the inclusion of

multiple training layers on domain-specific datasets whilst maintaining a sentence-level SA approach. This creates a trade-off between deeper understanding through LSTM functionality whilst maintaining a sentence-level granularity, unlike the aspect-level technology present in ABSA. This trade-off was likely to provide interesting results at the comparison stage, and as such the approach was chosen for the project.

2.5 Rejected Approaches

Created within SCC at Lancaster University, the BELLA project (Moore & Rayson 2018) was of particular interest from the offset. Created for aspect-level sentiment analysis within the restaurant domain, the project is state of the art and still in constant development, which may have provided interesting insights within the evaluation, particularly when tested on cross-domain datasets. However, it was decided that given the short timescale for the project and the need to understand, adapt and implement the current version of the project, the time would be better spent implementing widely-used, easily-adaptable tools.

TextBlob was also mentioned within Chapter 2.4.1 as a possible implementation for rule-based SA. Though it has been found in the past to be incredibly quick at analysing text, research supported that the tool performed comparatively poorly when evaluated against human perceptions of sentiment for fictional text (Hall 2020). Due to this previous work, and the domain specificity of VADER being trained on Twitter data (a potential evaluation dataset within this project) it was decided that VADER would be the favoured approach.

3 Datasets

For the purposes of the evaluation, it was deemed important to be able to test the approaches on datasets that differed from the traditional review format, as this would give a wider view of the cross-domain applications of current technologies and allow for more varied conclusions to be drawn later in the report. The final datasets chosen, as well as rejected options, are outlined below.

3.1 IMDb Review Dataset

As described briefly in Chapter 2.3, the IMDb review dataset (Maas et al. 2011) consists of 50,000 reviews from IMDb, with no more than 30 reviews from any one movie in order to give a broad area of topics and genres. The dataset has an even number of positive and negative reviews, denoting that any system randomly guessing polarity should yield a 50% accuracy. Review classification is completed using the inbuilt IMDb score, with a negative review denoted by a score of less than 5, and a positive review denoted by a score of more than 6. The problems of bias raised through this approach were outlined in Chapter 2.3. The dataset was chosen for evaluation as it is still used as a benchmark when testing state-of-the-art approaches to SA[9], allowing for it to act as a good "baseline" within the project. It will demonstrate the strength of the chosen approaches in a common domain before we branch out to the less traditional datasets. It is also easily available for download[10], and is pre-labelled with corresponding polarity values, making it usable for supervised training of models and to gather an accuracy score when evaluated against.

3.2 Friends Review Dataset

Whilst the IMDb review dataset provides a good opportunity to gather an accuracy score for the number of correct review polarities predicted, it was discussed in Chapter 1.1 that a key motivation of the project is examining how tools deal with the deeper nuance of a text. We want to evaluate more than just a binary "positive" or "negative" in terms of polarity, but also the strength of the assertion being made. For example, the sentence "Tom isn't a great person" would yield the same binary classification as "I hate Tom", although the strength of the opinion is drastically different. Capturing this difference in opinion strength in a way that allows for a comparative analysis of the tools was deemed essential within the report. Further to the point raised in Chapter 1.1 surrounding tracking the perceived sentiment of a character throughout the course of a work of fiction, the Friends Review dataset was collated in order to track the changes and strength of opinions over time.

The dataset consists of every IMDb review for every individual episode of Friends, with 1400 entries in total. The data was acquired using the Selenium[11] web-driver, which allows users to programatically scrape the HTML of a page at any given URL. The review text was then extracted from the HTML and saved into a txt format, with labels of the specific season and episode number to document the point during the show at which the opinion was true. The average review length is 275 characters, so there is a fairly large amount of text to analyse within the dataset. It could be argued, however, that users who are reviewing individual episodes are doing so after already having completed watching the entire show, and so are writing their review with a comparative context to the rest of the show as a whole. Considerations like these will be discussed further in the Results chapter of the report.

[9]Current SOTA available at https://paperswithcode.com/sota/sentiment-analysis-on-imdb
[10]IMDb review dataset available at https://ai.stanford.edu/ amaas/data/sentiment/
[11]Selenium Browser Automation available at https://www.selenium.dev/

3.3 Friends Scripts Dataset

Branching drastically away from traditional "review" formats for sentiment analysis, the idea of using a written script format for analysis of individual characters is an important step for cross-domain validation of the tools. In this context, even tools that have been built and trained with a specific movie review domain in mind will be tested on a completely novel dataset. The key features for comparison when evaluating tools on this dataset will be not only to examine how the outputs of the tools will differ from each other, but also how they differ from the sentiments extracted through the episode reviews dataset. It is important to explore whether extracting analyses directly from the source text will give a greater deal of nuance and context then when formulating sentiment from an indirect review of the same content.

The dataset consists of the scripts for every episode of *Friends*, available for free use online[12]. In a similar method to that explained in Chapter 3.2.2, Selenium is used to programatically access the script for an episode at a given URL, before extracting the text and saving to a text file. Pre-processing for the removal of stage directions and other extraneous data is outlined in Chapter 3.3. The usage of a locator tag for each episode also allows for the sentiment of an entity to be tracked over time, which allows for excellent comparative analysis when plotted on a sentiment graph.

3.4 Twitter Review Dataset

As mentioned in Chapter 1.1, work completed during fulfilment of the SCC.419 Industrial Placement module led to the creation of a rudimentary Twitter-scraper that can be used to find opinions related to specific hashtags. Within the context of the project, the ideal use for this tool was to collate a dataset of opinions expressed about any of the six main characters in *Friends*, so that the language used could be evaluated using the SA tools. This was deemed an interesting area of research as it would take the cross-domain evaluation in a new direction - analysing sentiment expressed in the unfiltered, 280-character, public realm of Twitter.

Within the first few weeks of the project, collection of Twitter data proved problematic, as it quickly became apparent that there was no way to verify at which point during their viewing of *Friends* a person had decided to Tweet an opinion about it. As such, it would be incredibly difficult to draw a comparative graph of how Twitter sentiments of characters change over time, and furthermore how each tool had evaluated these changes. Another challenge quickly presented itself as it became clear that the vast amount of opinions related to *Friends* on Twitter were not expressed explicitly as text, but were instead shown as images or GIFs.

Figure 5 shows an example of a Tweet returned when scraping with the search term 'Ross'. The SA approaches within this report have no way to extract meaningful insights from this data, as they work on a purely text basis. It also stands to reason that even if the images present were described as text, a SA tool would struggle to identify the point in the show at which each shown event occurs, as it requires a deeper understanding of the context of a scene than is currently plausible. Though these issues do highlight a potential future avenue of research through a cross-discipline study of Computer-Vision and SA for Twitter analysis, it fell outside the remit of the project. Due to these significant difficulties, the concept of a Twitter review dataset was discontinued early on in the project, though it remains an interesting topic for future work.

[12]Friends anthology scripts available at https://fangj.github.io/friends/

Figure 5: Example Tweet for Twitter Review Dataset

3.5 Table Of Features

Below is a summary table containing features for the three datasets used within the evaluation section of the report.

	Labelled?	Entries?	Average Length?	Domain?
IMDb Review Dataset	Yes	50,000	303 characters	Movie Review
***Friends* Review Dataset**	No	1,400	275 characters	TV Review
***Friends* Script Dataset**	No	165	20,305 characters	Fictional Script

Figure 6: Table showing the features of each dataset

3.6 Data Pre-processing

The preparation of data through pre-processing and cleaning is a fundamental step in carrying out effective, accurate NLP analysis. All datasets were cleaned using the same pre-processing pipeline, so that the tools could be applied and measured for an accurate evaluation. It is worth noting that the *Friends* script dataset was subject to additional data cleaning in the form of removing stage directions, writing credits, scene changes etc.

3.6.1 Tokenisation

The process of splitting a given text into individual tokens, tokenisation is vital for all aspects of NLP. A 'token' can refer to words, numbers, punctuation and numerous other components of written text. It is often the first step in the pre-processing pipeline, with many later tasks (lemmatization, NER, etc) accepting tokens as input data, rather than a string of text. It is a particularly useful function when calculating the frequency of particular tokens within the text, or when trying to match a token to an external lexicon in stop-word removal. Within the

project, tokenisation is completed on all datasets using NLTK's tokenise library[13], which can break down an input text into sentences and individual tokens.

3.6.2 Stop-Word Removal

Stop words are the most common words within a language, and can be removed from a text without changing the overall meaning. Stop word removal is a vital step in NLP, as the amount of processing time saved by reducing the overall number of tokens within a text can be enormous when evaluating large datasets. VADER employs the spaCy[14] lexicon of stop-words, and removes them automatically when processing data. Because of this, the spaCy stop-word list was removed from each dataset in advance, so that all tools could evaluate the same text. This was completed by programatically iterating through every token in each dataset and comparing it against the list of 326 stop-words in the spaCy lexicon, removing it from the dataset if a match exists. Figure 7 shows an example of stop-word removal. You can clearly see that the overall meaning of the sentence has not been lost by removing stop-words, but the number of tokens to be evaluated at a later stage has decreased by 40%.

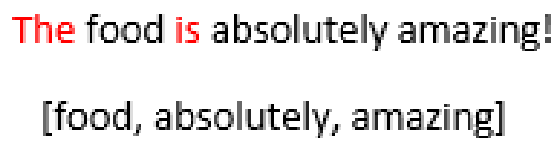

Figure 7: Example of stop-word removal

3.6.3 Lemmatization

The goal of lemmatization is to find the normalised form of a word. This means to reduce the inflectional forms and derivationally related forms of a word to a common base form (e.g., "am", "are", "is" would all be lemmatized to "be"). It is often associated with the process of stemming - where suffixes are removed from a word to obtain a root word (e.g., "intelligence", "intelligent", "intelligently" would all stem to "intelligen"). The problem with stemming is that it can produce intermediate representations of words that lack meaning (intelligen alone is not a word). This raises particular issues when using lexicon-based SA tools, which may not have an associated sentiment value for stemmed words, and will instead mark them as neutral. Lemmatization takes more time than stemming but produces word representations with meaning. As such, lemmatization is the second step of pre-processing for our datasets.

3.6.4 Named-Entity Recognition

A crucial part of aspect-target based SA solutions, Named-Entity Recognition (NER) is the process of locating and classifying named entities within a text. Used in our ABSA implementation, entities can be categorised into pre-defined categories such as names, organisations, locations, quantities, etc. Within ABSA, NER allows us to quickly identify potential aspect-targets amongst a corpus of verbs, adjectives and other non-noun tokens. NER struggles when identifying named-entities in fictional domains, where references made to characters and places in a fantasy world have no grounding in real-world tokens and locations. However, due to the context of *Friends* taking place in the very real New York City without any references to

[13]Word and sentence tokenisation available at https://www.nltk.org/_modules/nltk/tokenize.html
[14]spaCy - Industrial strength NLP capabilities available at https://spacy.io/

high fantasy, it is reasonable to expect that NER systems should show strong capability in our chosen domains.

Previous work indicates that of the two main libraries for NER (NLTK and spaCy), spaCy has a much larger lexicon of known proper nouns (Hall 2020) and features a better algorithm for spotting unknown entities through capital letters and the placement of tokens within a text. As such, it is advantageous that ABSA features spaCy NER as part of its implementation pipeline.

4 Methodology

This chapter outlines the justification for the overall development approach of the research, before focusing on the development of the experimental test-harness for each dataset, including how comparison metrics were collected for each tool.

4.1 Python

Python is a language with excellent functionality for data processing (Sanner et al. 1999), as well as being incredibly intuitive to use. Krill (2019) denotes that the rapid increase in popularity for the language (now second most-popular in the world) is due to widespread growth in the data science field. For the aims of the project, Python's strength in general purpose programming combined with its vast library support has made it an excellent option for data analysis (McKinney 2012). In the realm of NLP, Python has a large catalogue of libraries and functions built specifically to aid textual analysis, including the Natural Language ToolKit, Tensorflow and individual, open source implementations of many common SA tools.

4.2 Evaluation Methods

This section outlines the metrics that were chosen for use during the comparative analysis stage of the report.

4.2.1 IMDb Dataset Prediction Accuracy

For this experiment, the IMDb dataset of 50,000 reviews, split evenly between positive and negative, was used as a baseline dataset. This then allowed for each tool to evaluate every review in the dataset and predict polarity with a binary classifier. As mentioned previously, any system 'guessing' at polarity should receive a score of around 50%, and not even state-of-the-art SA systems have scored higher than 98% due to inherent flaws in the dataset (see Chapter 3.1). This evaluation metric gives us a good idea of how tools can evaluate sentiment in a traditional domain that would fit a likely use case in commercial implementations of SA. Three scores will be collected for analysis: percentage of correctly identified positive reviews, percentage of correctly identified negative reviews and average accuracy percentage. It also allows for an easy quantitative analysis without examining qualitative examples of each tool. It is worth noting that the Keras RNN implementation was trained specifically on the IMDb dataset, so we should expect this approach to generate a high accuracy score.

4.2.2 Time Taken

A second metric used in the evaluation of the approaches is the time taken to complete the above task. Information was collected using the time Python module on the IMDb dataset prediction task. Although the data used within the project is relatively small when compared with language models trained on millions of data points, it was deemed worth measuring the time taken for completion as this could provide a glimpse at how the tools may scale if applied to larger amounts of data. As the topic of environmental concerns is becoming increasingly prevalent when evaluating computationally expensive NLP/ML models (Strubell et al. (2019), Schwartz et al. (2019)), time complexity of large-scale data analysis systems could be seen as one of the deciding factors when choosing an NLP model in the coming years.

For the sake of clarity, the experimental testing was carried out on a 64-bit Microsoft Windows 10 PC operating an i7-6500U @ 2.5GHz processor, 8GB RAM, 256GB SSD (547Mb/s

read speeds measured using CrystalDiskMark 7[15]) compiling through Microsoft Visual Studio with limited background apps running.

4.2.3 Human Perceptions Questionnaire

As part of the wider scope of the project, it was deemed essential to explore how human perceptions of characters within the fictional domain would match up to the conclusions drawn by the various SA implementations. In order to achieve this, a group of 14 individuals were asked to fill out a questionnaire mapping their perceptions of the 6 main characters throughout the TV show *Friends*. The demographic of the group was university students aged 18-22 years old studying towards varying degrees from English Literature to Computer Science. This demographic was targeted as they were readily available and also covered a range of academic disciplines, which may help mitigate participants having similar approaches to fictional literature. A questionnaire was provided which asked participants to draw a graph of how they perceived the six main characters on various time scales: throughout all 10 seasons of the show and over the course of the final episode (participants were asked to rewatch the final episode before filling this section out). The graphs would demonstrate perceptions of characters from a scale of -10 to 10 pertaining to polarity, with -10 outlining a very negative view of a character and 10 outlining a positive view. The scale was kept to between 20 numbers so that participants would have enough scope to go into more than just "positive" or "negative", but is also kept small enough so trends are easy to be recognised. An example questionnaire has been included in the appendix **??**. It should be noted that ethics clearance was granted beforehand, in line with Lancaster University ethics guidelines[16][17]. The questionnaire asked participants to map their perceptions of:

- Chandler
- Monica
- Joey
- Phoebe
- Ross
- Rachel

All six of these characters were picked as they appear relatively consistently throughout the show, which allows for a better mapping of how perceptions change over time. For evaluation purposes, each tool will examine the sentiment of these characters over the same timescales, and will draw graphs of sentiment to allow for a qualitative, comparative analysis. If we assert that human-drawn perceptions are the best classifier for sentiment, then it follows that an approach that is able to replicate similar perceptions of sentiment should be the most advanced and accurate. This judgement is, in itself, subjective as people are often unable to agree on a specific scale or values for sentiment. This will be discussed in finer detail in Chapter 5.3.

It is important to note that the tools will be evaluated using the *Friends* review dataset. As this dataset contains at least 1400 references to each character, the use of a "chunker" is used in each implementation to group data points together and find an average before plotting

[15]available at https://www.majorgeeks.com/files/details/crystaldiskmark.html

[16]Lancaster University Science and Technology ethics guidelines presented at https://www.lancaster.ac.uk/sci-tech/research/ethics/

[17]Research privacy notice available at https://www.lancaster.ac.uk/research/participate-in-research/data-protection-for-research-participants/

the average on a graph, so as to reduce the number of data points on the output graphs. A chunker of size 70 will be used throughout, meaning that the first 70 sentences containing a search term will be averaged into a single data point and so on. With at least 1400 data points per experiment, a value of 70 was chosen to provide at least 20 data points per graph. With 10 seasons of content reviewed in the *Friends* review dataset, it was important to be able to visualise the overall trend, without being so reductive as to condense 24 episodes worth of reviews into a single data point for a season.

4.2.4 Individual Instance Examination

As well as overall trends in the data, it was also decided that the analysis should compare polarity rankings on an individual scale. This means that tools will be asked to evaluate a small dataset of individual sentences and provide a polarity score, which will be compared with human-perception scores of the same sentences. Human-perception scores were collected by a subset of 10 participants from the study above. Participants were asked to provide a value for their perceived sentiment of the sentences on the scale -1 to +1, in 0.1 intervals. Sentences were chosen to represent a range of complexity in terms of nuance and structure to push the limits of the SA tools. The sentences and average human sentiment (mean and standard deviation) are represented below:

- The food was amazing! - Mean 1, SD 0
- That film was awful - Mean -0.94, SD 0.09
- It was not the worst thing I had seen - Mean -0.23, SD 0.18
- You're killing it! - Mean 0.78, SD 0.27
- It had some really great parts but I did not like it overall - Mean -0.17, SD 0.22

4.3 Tool Implementations

The key aim of the project was to implement a way of comparatively evaluating SA tools when tested on traditional review corpora and cross-domain TV script formats. Due to the research-driven nature of the project, it was paramount to implement a test harness where results could be obtained quickly and fairly. As justified in Chapter 3.1, it was decided that the implementation would be built using Python3 as it is incredibly fast and supports libraries for all 3 tools under evaluation. The overall design consists of a user API, from which different experiments can be called. The system then presents experimental results to the user (accuracy scores, sentiment graphs, etc.). It was necessary to have a single API with multiple possible experimental calls so that all experiments would be carried out in the same environment on the same system in order to maintain accuracy, and so that experiments could be repeated easily to allow for quick collection of results. The ability to gather results quickly and accurately was pertinent given the 7 week timescale of the project. The following sections outline how each tool was implemented within the project.

4.3.1 VADER Implementation

VADER is available as a simple pip install when using Python. It has a straightforward pipeline when implemented - text is tokenised into sentences before each word is tokenised. A SentimentIntensityAnalyser object is created, which is used to provide a polarity score for each sentence being evaluated. For evaluation of individual entities within a dataset (e.g., for experiments looking to track perceptions of a character over time in the *Friends* review dataset),

a search of all data entries is carried out with the entity name as the identifier. All of the sentences containing the identifier are extracted, before being fed into VADER for evaluation.

The output of VADER for each evaluation text is four scores: positive, negative, neutral and compound scores. The first three scores are ratios for proportions of the text that fall into each category, and as such should all add up to one. These scores are useful when analysing how sentiment is conveyed in rhetoric for a given sentence, for example different writing styles may reflect an abundance of strongly worded rhetoric (yielding a low neutral score) where other styles may produce a high neutrality score whilst reflecting the same level of overall sentiment. It is important to note that these proportions represent the raw categorisation of each token, and do not account for VADER's inbuilt rule-based enhancements which account for punctuation amplifiers[18] and negation polarity switches[19].

As the rule-based modifiers mentioned above are important to present an idea for the overall classification capabilities of VADER, we will use the compound score as an output figure. This score is computed by summing the valence scores of each word in the lexicon, adjusting for the lexical rules and normalising between -1 and 1. This normalised score is particularly useful for comparison with other tools, as it is the same output scale as both Keras and ABSA.

4.3.2 ABSA Implementation

ABSA is an open-source Python library that can be directly accessed via a pip install. It is incredibly simple to use, as shown in Figure 8, which outlines how the ready-to-use NLP pipeline can be loaded in with two short lines of code. It is worth mentioning that different models can be loaded in at this stage (five are currently supported), but the default implementation uses an adaptation of BERT, as discussed in Chapter 2.4.2. This highlights a potential further avenue of research in evaluating the currently available implementations.

```
import aspect_based_sentiment_analysis as absa
nlp = absa.load()
```

Figure 8: Setting up ABSA for SA

Though the resulting NLP object is now ready to evaluate any text passed through it, it is advantageous to break down the steps contained within the pipeline in order to highlight the steps taken throughout.

```
18  model = absa.BertABSClassifier
19  tokeniser = absa.BertTokeniser
20  task = model.preprocess(text=..., aspects=...)
21  tokenised_examples = model.tokeniser(task.examples)
22  input_batch = model.encode(tokenised_examples)
23  output_batch = model.predict(input_batch)
24  predictions = model.review(tokenised_examples, output_batch)
25  completed_task = model.process(task, predictions)
```

Figure 9: Breakdown of ABSA pipeline

Figure 9 shows how the BERT Aspect-Based classifier is loaded into a model object. From here, the input text is broken down in line 20 into tasks. A task keeps an example, consisting of a pair made up of a sentence of text and an aspect. These examples are then tokenised and

[18]The use of punctuation to amplify the meaning of a sentence, i.e. using an exclamation point at the end of a phrase to convey excitement

[19]Detecting the presence of words like "not" before a strong polarity word in order to reverse the polarity

encoded before being passed to the model. The model then predicts the correct output for sentiment classification. Line 24 shows a review phase, whereby the model has the explain why it has predicted a sentiment in a particular way (this justification step is beyond the scope of the project and as such we are only interested in the resulting output polarity score).

The model can now be fed input texts and associated aspects. For the purposes of evaluating sentiment scores of individual characters, the character name will be fed into the model as an aspect. The aspect field will be left blank when evaluating the IMDb review dataset, so that all of the data will be analysed, rather than instance references to specific aspects.

4.3.3 Keras Implementation

Keras is implemented using Tensorflow[20] which is accessible with a pip install. It also has capabilities for downloading datasets easily, allowing for the labelled IMDb review dataset to be fed straight into the program. Whereas VADER works completely off the shelf, we have to build a Keras model which we can then use to evaluate text.

```
30  model = tf.keras.Sequential([encoder, tf.keras.layers.Embedding(input_dim=len(encoder.get_vocabulary()),
31                               output_dim=64, mask_zero=True),
32                               (tf.keras.layers.LSTM(64)),
33                               tf.keras.layers.Dense(64, activation = 'relu'),
34                               tf.keras.layers.Dense(1)])
```

Figure 10: Building the Keras model

Figure 10 shows how a sequential model of 64 LSTM layers is built. This means that for each line of text that is evaluated, it is processed 64 times with each iteration passing forward information that is has collated from the vocabulary of the entire dataset. The 'Dense' method on line 33 collates all of the previous information into a single output parameter of sentiment.

After building, the model is compiled and is ready to be fitted to data.

```
model.fit(train_data, epochs=10,
          validation_data=test_data,
          validation_steps=30)
```

Figure 11: Fitting the Keras model to IMDb dataset

Figure 11 shows how the model is fed the training data (a randomly selected 10,000 reviews from the IMDb dataset) before declaring the number of epochs (training iterations). The validation data is then set to another random 10,000 entries from the dataset, and the validation steps are set to 30 (values used in the benchmark Keras implementation that achieved an accuracy score of 89%). This means that for each epoch, each sentence is run through the model 30 times, with the highest result outputted. Due to this high number of iterations, the model takes a long time to train (2 hours and 27 minutes). Although this may seem like a disadvantage to the approach, especially as the training dataset is relatively small compared to larger language models, it is worth noting that this is a singular event after which the model can be used quickly and easily. Once training is completed, the model is saved and ready for usage on any other text.

[20]Tensorflow is an end-to-end open source platform for ML that contains many tools, libraries and resources

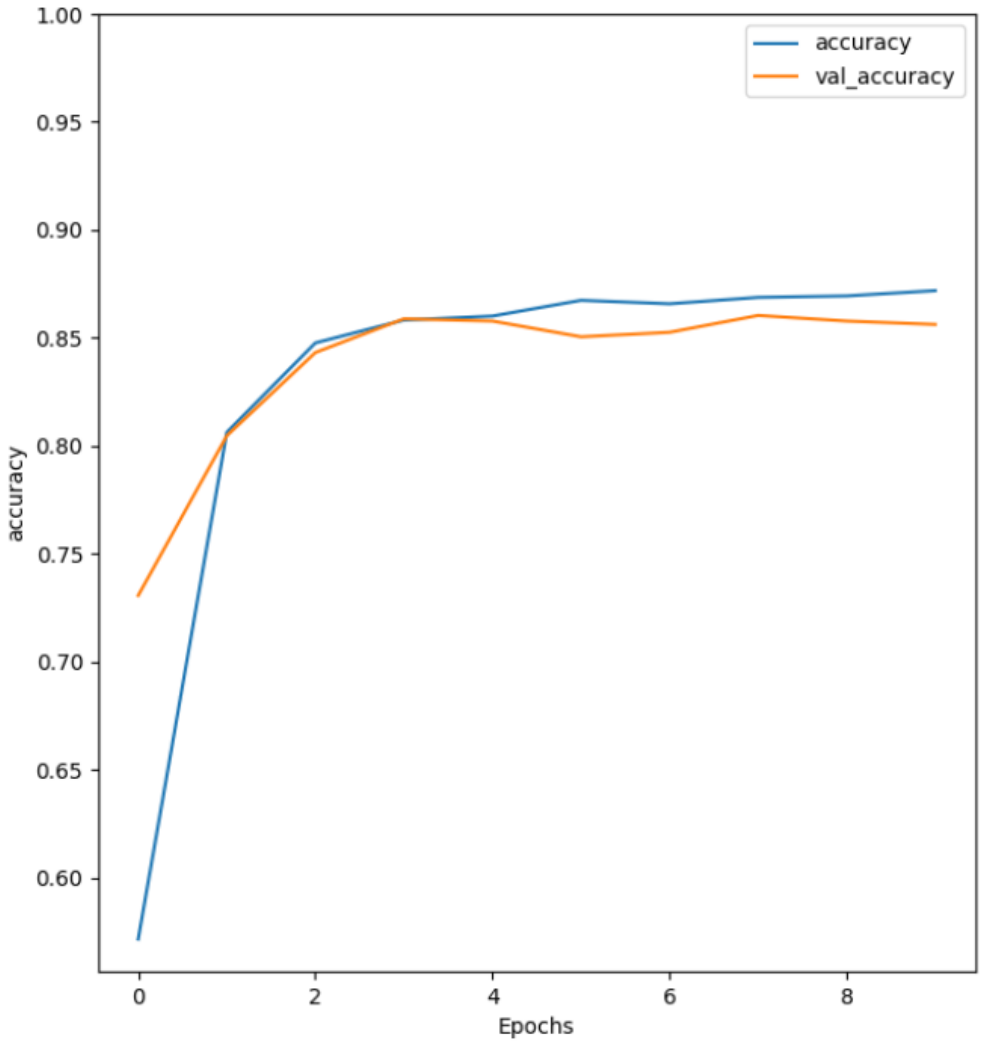

Figure 12: Training iterations of the Keras model on IMDb dataset

Figure 12 shows how the accuracy of the model increases after each epoch. Accuracy refers to the ratio of correctly predicted texts in the testing set (which features corpora of the same film review domain), whereas the val_accuracy is the accuracy ratio when using the model to predict an external corpus, provided by Tensorflow, that covers a range of domains (e.g. restaurant reviews, laptop reviews, hotel reviews). In light of this, it can generally be said that whilst accuracy refers to the performance of the model on this specific domain, val_accuracy gives an indication of how well the model would perform on an unseen dataset.

5 Results

This chapter will look at the collection, presentation and discussion of results obtained from each SA tool implementation for the evaluation metrics raised in Chapter 4.2.

5.1 Comparative IMDb Dataset Prediction Accuracy Scores

For this collection of results, all experiments were carried out on the IMDb review dataset. Each SA tool was asked to assign a polarity score of positive or negative to each of the 50,000 reviews present in the dataset before totalling the number of correctly guessed reviews (compared with labels present in the dataset already). The strength of the polarity is not of concern in this experiment - once each tool was normalised to provide an output between -1 and 1, any text scoring less than 0 was classified as a negative review, whilst any score over 1 was classified as positive. The experiment was completed three times for each tool, with the average accuracy presented in the table below. It is worth noting that the accuracy was consistent for each iteration of the experiment for both VADER and Keras. Having found this surprising, a full code-review took place before all experiments were redone, though the results remained the same. This can be explained to some degree by each run being completed on the same dataset (the full set of 50,000 reviews), so rule-based tools like VADER should evaluate the text in exactly the same way. For Keras, having been trained on a subset of the dataset, this results could be due to the model overfitting, which will be discussed later. Regarding ABSA, the variations in accuracy were miniscule (0.03% after a second repetition of the experiment) and could potentially be due to any number of small processing errors. However, this variation could prove that ABSA is very consistent in how it evaluates data, which proves beneficial for its reliability as a SA tool.

It should be noted that "accuracy" in this moment refers to the percentage of positive or negative reviews correctly identified as their appropriate polarity. For example, if a dataset containing 10 positive reviews was identified as containing 6 positive reviews by a model, this would yield a positive accuracy score of 60%.

	Pos Accuracy %	Neg Accuracy %	Avg Accuracy %
VADER	85.14%	53.63%	69.39%
ABSA	72.30%	73.11%	72.71%
Keras	77.15%	77.02%	77.09%

Figure 13: Average accuracy when tested on IMDb Review Dataset

Figure 13 highlights the average score for each tool when evaluating the polarity of the IMDb review dataset. A key conclusion to be drawn from these results is that the Keras RNN implementation achieved the highest score overall of 77.09%, although this is still significantly lower than the state-of-the-art scores that range from 80% up to 98%. This relatively low performance could be due to two key factors - poor tuning of the model (parameters such as the number of epochs and the size of the training/testing datasets could have been altered) and the possibility of over-fitting. Regarding the first explanation, the implementation used within the experiment uses the same parameters as presented in Yu et al. (2019), where an accuracy score of 89% was achieved on the same dataset. Due to this, overfitting is the more likely cause of our low score. Overfitting occurs when a model becomes so fine-tuned to a small training set that it starts to struggle when evaluating anything outside of this training data. As the Keras RNN model is trained on a randomly selected 10,000 reviews from the dataset, it is possible that the sample for this implementation did not reflect the dataset fully, and as

such the model struggled when evaluating every text within the corpus. A possible extension of this work could include training numerous models on different random subsets of the IMDb review dataset, in order to see if this problem persists. However, due to the large amount of time taken to train an RNN model, this will not be possible within the context of this project.

The ABSA implementation performed better than expected in the experiment. Although it is not reaching groundbreaking accuracy scores, the positive and negative recall rates are very similar and higher than VADER by comparison. It was the expectation that the aspect-based tool would struggle when there was a lack of a specific target to evaluate, but this does not appear to be the case. It is unclear how the model evaluates text in this situation - it is more than likely that the tool analysed each sentence individually and assigned sentiment scores to each before finding an average for the whole review. This behaviour would effectively reflect the methodology used by VADER, but with a more advanced training lexicon in the form of BERT_BASE.

The next conclusion to be drawn from the data presented in Figure 13 is the disparity between the positive and negative recall scores for the VADER implementation. Though the amount of correctly identified positive reviews is the highest for this tool, it also has the lowest negative recall rate within the analysis, at just over 50%. One possible explanation for this gap is the data used to train VADER. Built primarily to examine social media texts, it is possible that the lexicon for the rule-based approach contains significantly more positive terms than negative terms, causing the positive language to be picked up and evaluated more often. With a heavy focus on text that could be found on Twitter, the lexicon may contain more visceral negative terms (curse words, expletives, etc) and may rank these extremely negatively, whereas more subtle negative language (more likely to be used in a formal IMDb review) is closer to a neutral score in comparison. These results clearly show that VADER is the optimal choice when evaluating sentiment on a mainly positive dataset, though the likelihood of this as a use-case is debatable.

Although this was obviously a low-budget production, the performances and the songs in this movie are worth seeing. One of Walken's few musical roles to date. (he is a marvellous dancer and singer and he demonstrates his acrobatic skills as well - watch for the cartwheel!) Also starring Jason Connery. A great children's story and very likable characters.

Figure 14: Example positive review from IMDb review dataset

Outlandish premise that rates low on plausibility and unfortunately also struggles feebly to raise laughs or interest. Only Hawn's well-known charm allows it to skate by on very thin ice. Goldie's gotta be a contender for an actress who's done so much in her career with very little quality material at her disposal...

Figure 15: Example negative review from IMDb review dataset

	Pos Review	Neg Review
VADER	0.91	0.53
Keras	0.55	-0.80
ABSA	0.27	-0.87

Figure 16: Polarity scores for example reviews

Figures 14 and 15 show examples reviews from the IMDb reviews dataset. Figure 16 shows the polarity scores assigned to these reviews by each tool. VADER has the worst performance here, as it incorrectly identifies the negative review as a strong positive polarity. It is intriguing to try and understand the tokens within the review that generated this positive sentiment, as the presence of words such as "outlandish", "unfortunately" and "struggles" would all hint at a negative perception, especially when exacerbated by the lack positive tokens (arguable "laughs" and "charm"). This role is reversed when we examine ABSA, which generates a low positive polarity score for the positive review, which contains phrases such as "very likeable" without much text that could be perceived as negative. Furthermore, these tendencies towards a more negative output for ABSA and a more positive output for VADER align with the results in Chapter 5.3.

5.2 Comparison of Time Taken

As mentioned previously, the time taken for each tool to evaluate the IMDb dataset in the above experiment was measured using the Python time library. The time was calculated from the moment the first review started being analysed to the end of the final review, skipping the steps of loading in models and datasets. The results are shown below.

	VADER	ABSA	Keras
Attempt 1	253	778	953
Attempt 2	90	807	1035
Attempt 3	60	706	932
Avg Time (Seconds)	134.3	763.7	973.3

Figure 17: Time taken for each tool to evaluate IMDb review dataset

When compared with the results presented in Chapter 5.1, Figure 17 shows a clear correlation between the time taken and the overall accuracy of the approach. VADER takes a fifth of

the time required by ABSA to complete the experiment, whilst only accounting for a 3% loss in accuracy. This trade-off becomes increasingly severe if we consider an evaluation of much larger datasets, where this time-gap becomes a major drawback of both the ABSA and Keras approaches. There is a simple explanation for the impressive speed of VADER, relating to time complexities as outlined below.

- VADER: O(n)
- ABSA: O(n!)
- Keras: O(n)

With VADER, performing a search for each word within the lexicon is a simple procedure that has a linear time complexity relative to the length of the text being analysed. Meanwhile, ABSA creates permutations of every text it evaluates in order to understand the relationships between aspects in a string (i.e. for the string "the cat sits", ABSA would evaluate "the cat sits", "the sits cat", "sits the cat" etc), which can be described as a factorial time complexity. This means that as the length of a sentence grows, it takes increasingly long to compute the result. Finally, Keras is still a sentence level tool that operates on a set number of iterations. This means that although the time complexity is linear (the size of an input string directly correlates to the time taken), the base function still takes longer as each input is passed through 64 LSTM layers. Sak et al. (2014) find that the time complexity of an LSTM model is directly correlated to the number of underlying parameters of the model, rather than the size of the input string. Therefore, with much larger input strings, it is likely that Keras would drastically outperform ABSA in terms of speed.

It is also worth noting that the time taken for VADER to complete in this instance drops dramatically after each attempt, taking less than a quarter of the time by the final iteration. This drop-off in time has proved incredibly difficult to explain. After extensive research, the VADER documentation does not appear to mention a caching function of any kind, which would in theory speed up the process at runtime. However, having completed the experiment again with a further three iterations, the results persist. It is worth noting that when running three separate instances of the test harness in different programs, the time taken was then consistent with values provided on the first attempt. Although there is potential that this phenomenon is caused by a built in feature of the Visual Studio Code development suite, these results do not persist for the ABSA and Keras implementations, where the time taken fluctuates. On the other hand, if VADER does have a form of caching algorithm, this could be incredibly beneficial when evaluating large amounts of data multiple times, as the potential time saved on each iteration increases the processing speed of what is already the fastest SA tool.

Finally, it is of note that the time taken for all three approaches varies wildly. Referring back to Chapter 5.1 with a particular focus on ABSA and Keras, we can see that although the models were incredibly consistent with their accuracy scores across multiple attempts, the time taken to perform the experiment can vary by as much as 10%. These inconsistencies are again worth considering on a larger scale, where a 10% variation in time could equate to hundreds of hours and vast amounts of computing power.

5.3 Comparison to Human Perception Graphs

In this experiment, all 3 tools were used to evaluate the sentiment of 6 entities within the *Friends* review dataset. Each review contains an episode locator, which allows for the results to be plotted on a graph in order to show how sentiment changes as the episodes advance. As mentioned in Chapter 4.2.3, each implementation uses a "chunker" of size 70 to group together data points into a single point, so that a trend line can still be seen on the output graph.

It is worth noting that the use of a chunker condenses the results around the neutral point. Although essential to the results (without it, the graph output is a dense block of lines plotting between 1400+ reviews), the averaging of data points pulls the line of best fit towards zero, as positive and negative scores are combined into a single point. Considering this, all data points have been multiplied by two in order to maintain the trend-line whilst fitting the data to the axes.

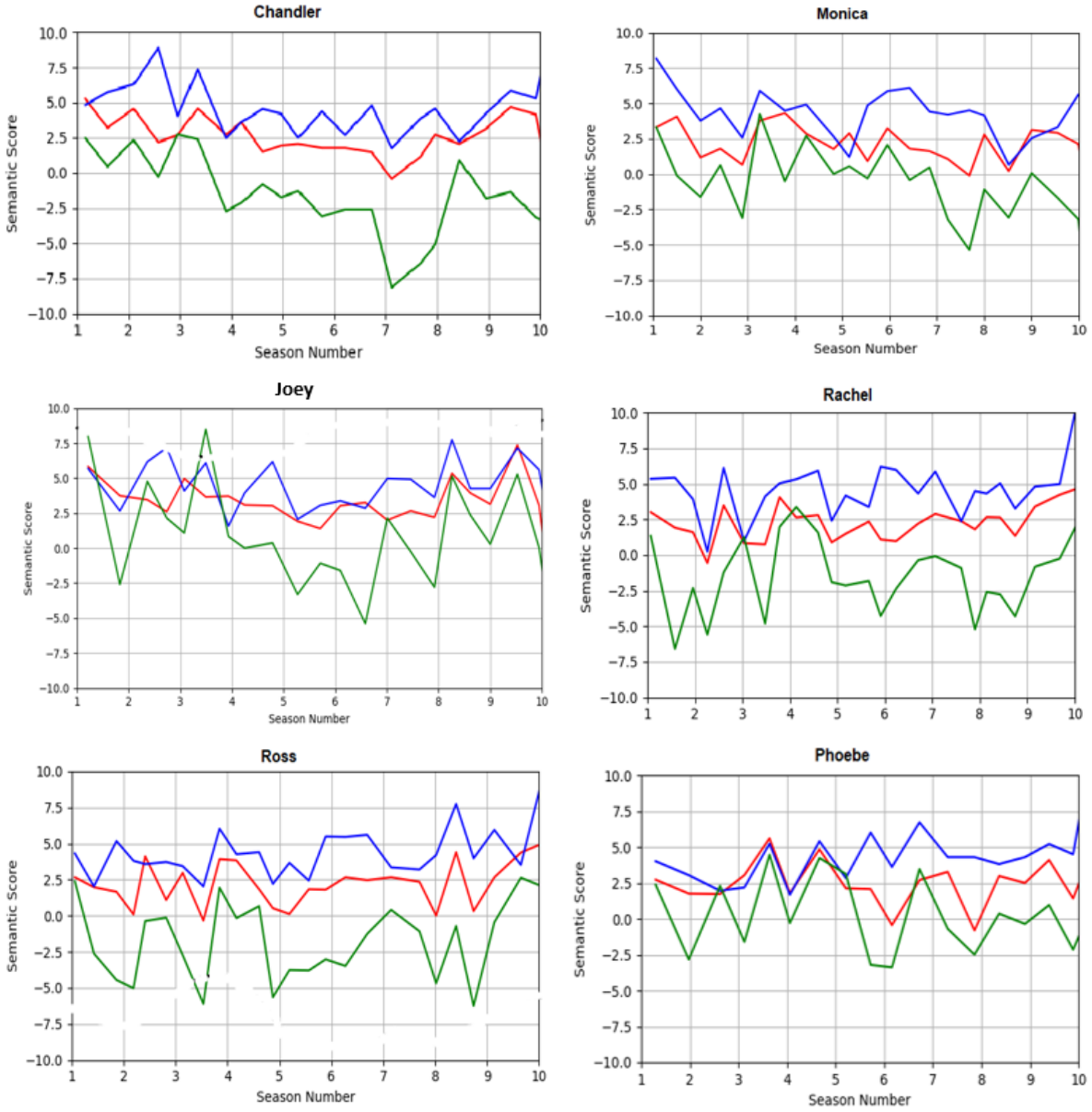

Figure 18: SA Tool output graphs of every character

The first key finding from the results is that the three models appear to follow similar trend lines, but begin at different polarity scores. Figure 18 shows considerable similarities between the shape of the trend lines, even overlapping and generating the same polarity score at various points. This could be due to similar polarity classifications for certain words. For example, the VADER lexicon and Keras model may assign the word "awesome" the same sentiment score, regardless of whether it was decided through a pre-defined lexicon or iterated training.

Furthermore, it is clear from Figure 19 that ABSA appears to have an overall negative perception of Chandler throughout the show, however this does not align with the mean human

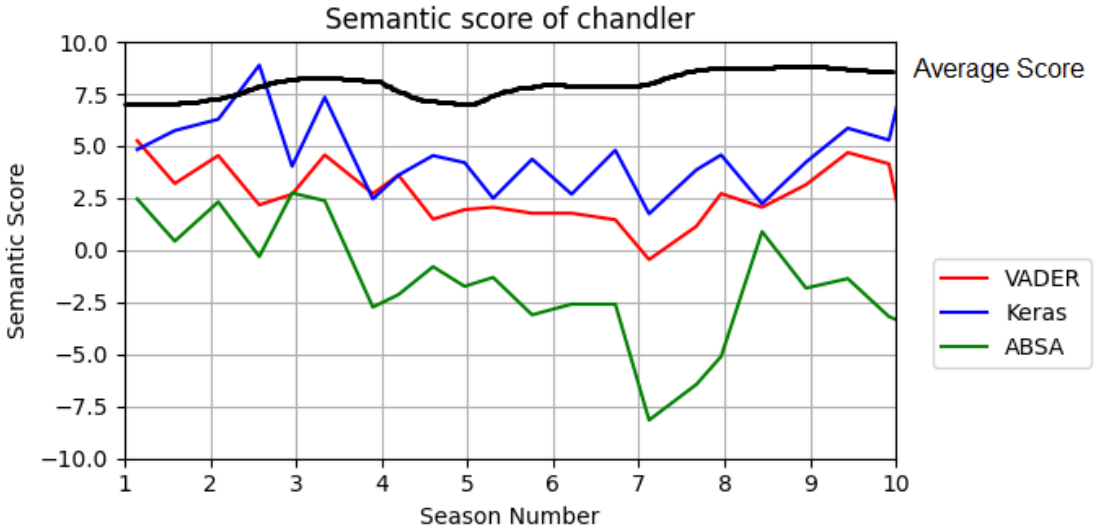

Figure 19: Human perception graph of Chandler

perception of the character. On the other hand, Keras seems to be closest to the human perception, with scores of at least 2.5 throughout. This disparity is clear at Season 7, where there is a full 10 score gap between the perceptions of ABSA and Keras. The strengths of Keras in this instance can be attributed to its domain training. Having been trained on the IMDb movie review dataset, the model is already familiar with the type of language present in online media reviews, and as such is able to correctly classify sentiments expressed in similar ways. This can also be used to explain the strong performance of VADER (remaining in positive polarity throughout); as a tool specifically designed to measure sentiment in online text, VADER is more fine-tuned to recognise and assign higher sentiment to language of this domain. It also follows that fans of the TV show *Friends* are more likely to take the time and effort to write a review for an individual episode, and so it is probable that as a whole the dataset tends towards positive polarity, which makes the resulting ABSA classifications interesting.

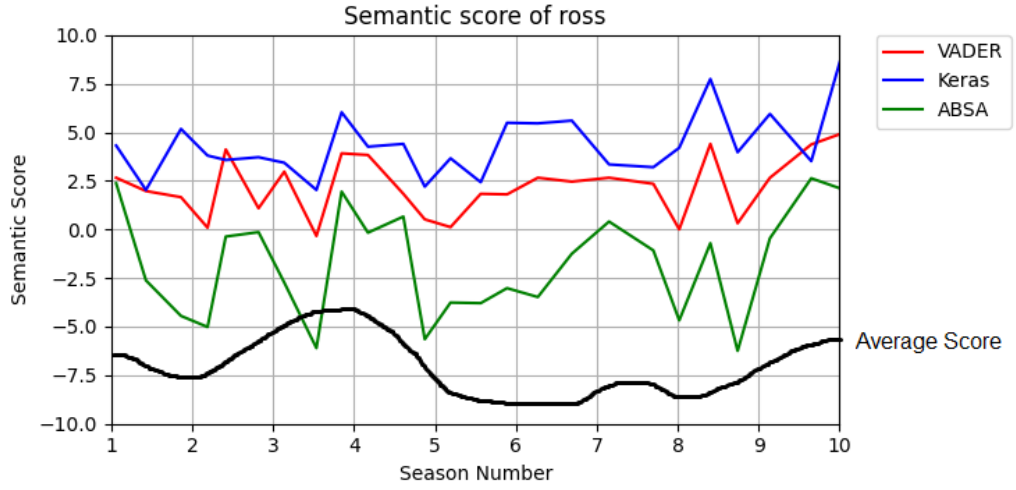

Figure 20: Human perception graph of Ross

Controversially, Figure 20 shows that these trends of polarity maintain even when evaluated against what many perceive to be a negative character. The data follows the same pattern - Keras is very positive, VADER is quite positive and ABSA is majority negative - though the average human perceived sentiment score in this instance across all participants is strongly negative. This potentially highlights a key issue with comparing SA systems and human perceptions; where SA tools can only evaluate an entity based on the data put before them, we have wider context on aspects of fictional work. For example, many articles have been released in recent years highlighting that Ross' behaviour throughout the show is unacceptable according to the social dynamics of the modern day, but the IMDb reviews within the dataset were mainly written between 2005 and 2015. This brings into discussion the topic of whether SA tools can ever truly replicate human perceptions, because there are a number of factors outside of the text that affect a person's opinion. This will be analysed further in the next experiment.

5.4 Comparison of Individual Instances

Subjectivity in SA has been a concern since the very emergence of NLP techniques, with papers written on how to combat the topic falling short of a full solution (Montoyo et al. 2012). As presented in Chapter 4.2.4, a group of 10 people struggled to evaluate five relatively simple sentences with any degree of unanimity.

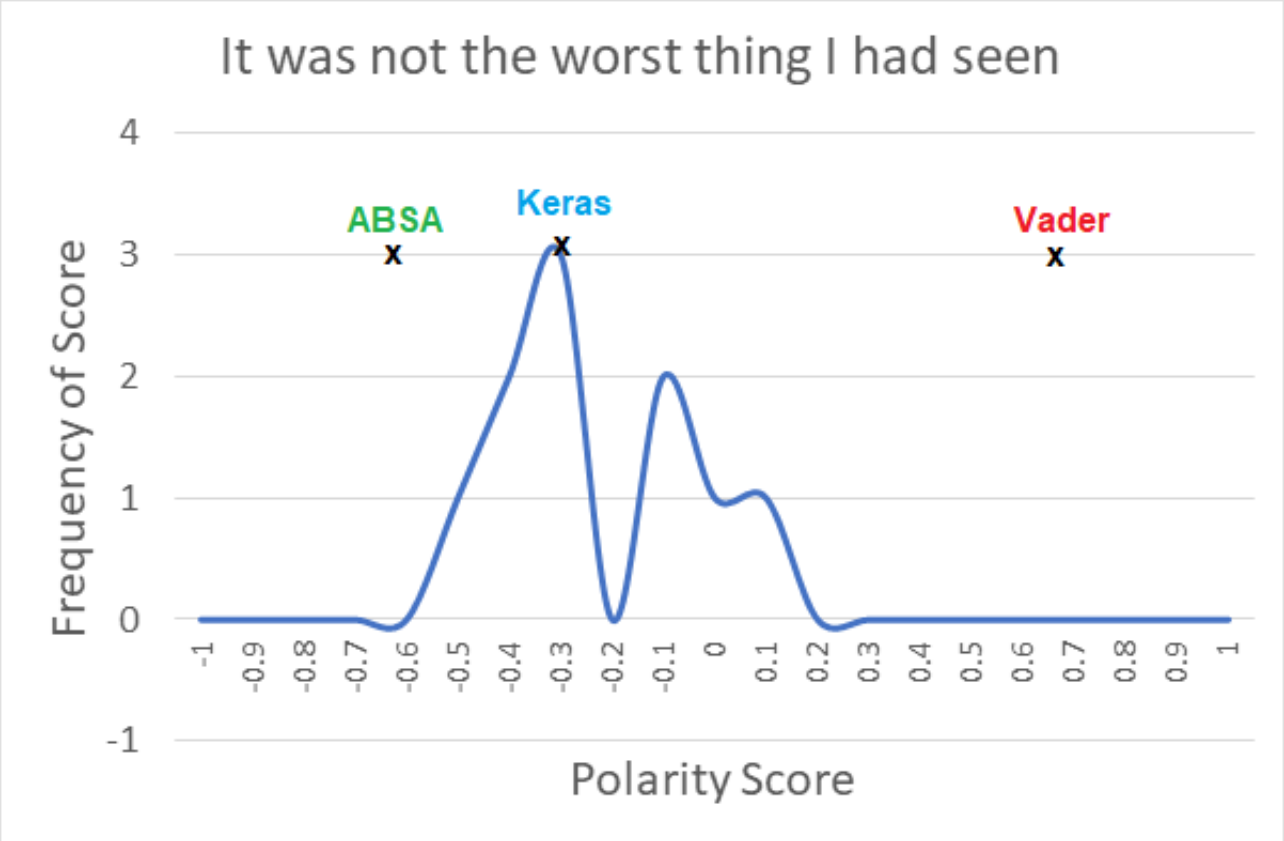

Figure 21: Frequency of human perception polarity scores with tool evaluations

Figure 21 shows the results of evaluating the sentence "it was not the worst thing I had seen" with both human sentiment and the SA approaches. It demonstrates clearly that not only do the SA tools struggle to agree with each other when analysing the sentiment of a text, but the human perceptions themselves struggle to reach a conclusive agreement. Though the Keras implementation accurately predicts the most frequent perception of this text, this is a relatively small sample size that could just as easily have perceived the polarity as positive.

Highlighted in a 2018 YouGov poll[21], this subjective nature of sentiment only emphasises how difficult SA is as a discipline, and helps to explain why state-of-the-art SA models have never breached over 98% accuracy (some researchers remain skeptical that it ever will).

5.5 A note on the Friends Scripts Dataset

It was mentioned early in the project that it would be beneficial to evaluate the SA tools on in an nontraditional domain and format. Due to this, it was decided that the tools should look to draw sentiment graphs of characters over time through the show, similarly to the experiment conducted in Chapter 5.3.

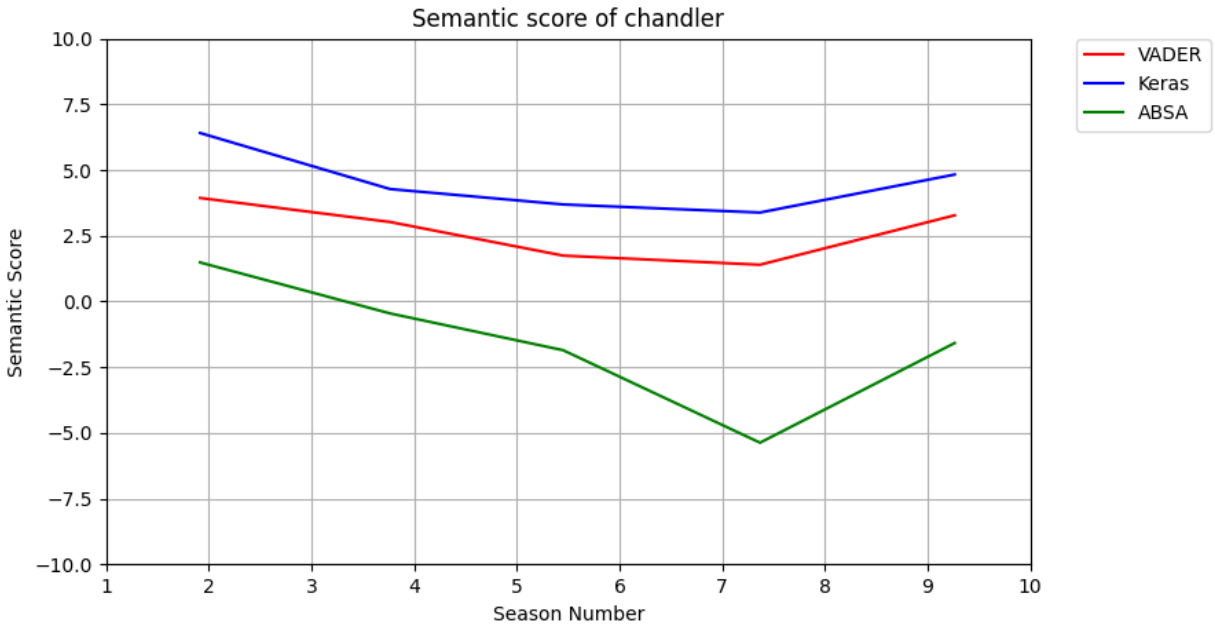

Figure 22: Output of running SA on "Chandler" with Friends Scripts dataset

However, Figure 22 clearly demonstrates how SA tools struggle to evaluate text in this domain. The lack of many data points can likely be explained by the format of a fictional script itself - the majority of text in the scripts is dialogue spoken by the characters. As such, the SA tools do not evaluate text spoken by a character themselves, but only react when characters are referenced within the dialogue (which usually occurs in the Friends scripts as an exclamation clause, e.g. "Joey!", which in itself does not give a good indication of sentiment.

Although it would be undoubtedly interesting to create a method for evaluating the sentiment in dialogue in written scripts (e.g analysing every line of dialogue a character says before plotting a graph), it is debatable whether this would provide a good representation of sentiment towards the character, or rather their expressions of sentiment towards other entities. This question could well be an entire research paper in itself and so falls outside the scope of this project, but it is an exciting potential avenue of work.

[21] *How Good Is "Good"? This Is How People Rated Adjectives From Best To Worst* available at https://www.iflscience.com/editors-blog/how-good-is-good-this-is-how-people-rated-adjectives-from-best-to-worst/

6 Conclusions

Through the completion of complementary human and machine based SA experiments which provided results for both a quantitative and qualitative analysis, we were able to obtain a number of key findings. These are outlined below as well as an evaluation of how the project aims were developed and met. A self reflection of the project process as a whole will be given before closing remarks will be made with reference to wider discussions.

6.1 Key Findings

	Avg Accuracy %	Avg Time (Seconds)	Time Complexity
VADER	69.39%	134.3	O(n)
ABSA	72.71%	763.7	O(n!)
Keras	77.09%	973.3	O(n)

Figure 23: Quantitative experiment results

Figure 23 outlines the quantitative findings of the project, with best performing approaches highlighted in green. From this alone, it is clear to see the trade-off between time taken and accuracy for each tool. It could be asserted that VADER is the best approach when it is necessary to evaluate text quickly with a reasonable level of accuracy, although the individual reviews presented in Chapter 5.1 show how VADER has a tendency to incorrectly gauge sentiment as positive. On the other hand, ABSA tends to evaluate sentiment overly negatively, as shown in Figure 16 and throughout Chapter 5.1, when it routinely gave an output of sentiment below zero, even when drastically contradicting the human-perception findings. Keras appears to be a good middleground between the two; though creation of the model is time-consuming, the fine-tuning of parameters and training stages of the model make the approach ideal for domain-specific tasks. In terms of cross-domain accuracy, the project was let down by challenging representations of fictional datasets.

6.2 Reflections on Aims

1. *Provide a detailed overview of current Sentiment Analysis methods, including strengths and shortcomings, as well as how these approaches have developed over time.*
   This was covered during the background chapter of the report to a sufficient level of detail relevant to the rest of the project.

2. *Identify and explain individual tools for each type of method which can then be implemented for testing.*
   I was able to identify and explain 3 key approaches to SA (VADER, ABSA and Keras). I also identified more methods and was able to justified why I had chosen not evaluate them within the scope of this project.

3. *Evaluate each of these tools against a large IMDb movie reviews dataset, individual IMDb reviews and the full anthology of scripts from the TV show Friends*
   I have completed an evaluation of each tool across all three datasets. I believe that having discussed all experimental findings I have given adequate suggestions regarding the best tool to use for specific tasks.

4. *Explore how the tools perceive opinion when comparing human perception graphs of sentiment with opinions expressed on Twitter, to gain an understanding of how they evaluate subtlety.*
   Unfortunately, I was unable to collect a dataset of Twitter reviews for *Friends*, though this does highlight a future avenue of work.

5. *Explore the reasoning behind the conclusions drawn through testing before making inferences on the best use cases for each method and how the tools could be improved in future.*
   I have explored the reasoning behind each key finding that arose in the Results chapter. When evaluating tools, I suggested potential use cases and future improvements for many approaches.

6. *Make references to how this information can be best utilised within the wider field of NLP*
   I have completed this as part of the Future Work section within this chapter.

6.3 Future Work

Although the findings of this report may not be groundbreaking in the field of SA by themselves, the project has emphasised the difficulty of SA whilst highlighting future avenues of research. A cross-discipline investigation into Computer Vision and SA for analysing Tweets online would be an intriguing line of work, as this novel way of expressing sentiment becomes increasingly prevalent in modern life. The optimisation of a Keras model through training on different subsets of data would also be interesting to see, as it is believed that a fine-tuned model trained on labelled data in the fictional domain (possibly in a script) could push the boundaries of modern SA into a new, exciting field entirely.

6.4 Closing Remarks

Over the course of the 7 week dissertation, I have implemented a research driven approach for the evaluation of three SA technologies. Overall, I am very proud of the amount of work that I was able to complete in such a small time-frame, and in a difficult time of my life. I thoroughly enjoyed diving into the world of sentiment analysis and now consider myself incredibly knowledgeable on the subject. With that being said, I do now have a greater appreciation for the complexity of these sentiment systems. I feel that I struggled to implement some cutting-edge technologies that were part of the original project focus, and fell behind on time because of this. On the other hand, I feel that the work that I have carried out is beneficial to the field of SA as a whole, and has opened up many further avenues of work for the future. On the whole, I complete this dissertation proud, well informed and eager to to see what the future of sentiment analysis holds.

References

Al-Muzaini, H. A., Al-Yahya, T. N. & Benhidour, H. (2018), 'Automatic arabic image captioning using rnn-lst m-based language model and cnn', *International Journal of Advanced Computer Science and Applications* **9**(6).

Asghar, M. Z., Khan, A., Ahmad, S., Qasim, M. & Khan, I. A. (2017), 'Lexicon-enhanced sentiment analysis framework using rule-based classification scheme', *PLOS ONE* **12**(2), 1–22.
URL: *https://doi.org/10.1371/journal.pone.0171649*

Behdenna, S., Barigou, F. & Belalem, G. (2018), 'Document level sentiment analysis: A survey', *EAI Endorsed Transactions on Context-aware Systems and Applications* **4**(13).

Bhutani, B., Rastogi, N., Sehgal, P. & Purwar, A. (2019), Fake news detection using sentiment analysis, *in* '2019 Twelfth International Conference on Contemporary Computing (IC3)', IEEE, pp. 1–5.

Chen, S., Peng, C., Cai, L. & Guo, L. (2018), A deep neural network model for target-based sentiment analysis, *in* '2018 international joint conference on neural networks (IJCNN)', IEEE, pp. 1–7.

Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M. & Ghosh, R. (2013), Exploiting domain knowledge in aspect extraction, *in* 'Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing', pp. 1655–1667.

Choi, G., Oh, S. & Kim, H. (2020), 'Improving document-level sentiment classification using importance of sentences', *Entropy* **22**(12), 1336.
URL: *http://dx.doi.org/10.3390/e22121336*

Ciravegna, D. et al. (2001), 'Adaptive information extraction from text by rule induction and generalisation'.

Delavenay, E. & Delavenay, K. M. (1960), *An introduction to machine translation*, Thames and Hudson London.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019), 'Bert: Pre-training of deep bidirectional transformers for language understanding'.

Dror, R., Baumer, G., Shlomov, S. & Reichart, R. (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)', pp. 1383–1392.

Elbagir, S. & Yang, J. (2019), Twitter sentiment analysis using natural language toolkit and vader sentiment, *in* 'Proceedings of the International MultiConference of Engineers and Computer Scientists', Vol. 122, p. 16.

Gao, Z., Feng, A., Song, X. & Wu, X. (2019), 'Target-dependent sentiment classification with bert', *IEEE Access* **7**, 154290–154299.

Ghiassi, M. & Lee, S. (2018), 'A domain transferable lexicon set for twitter sentiment analysis using a supervised machine learning approach', *Expert Systems with Applications* **106**, 197–216.

Hahn, U. & Mani, I. (2000), 'The challenges of automatic summarization', *Computer* **33**(11), 29–36.

Hall, O. (2020), 'Evaluating natural language processing methods as a tool for geographical text analysis within both historical and fictional corpora'.

Hasan, A., Moin, S., Karim, A. & Shamshirband, S. (2018), 'Machine learning-based sentiment analysis for twitter accounts', *Mathematical and Computational Applications* **23**(1), 11.

Hermanto, A., Adji, T. B. & Setiawan, N. A. (2015), Recurrent neural network language model for english-indonesian machine translation: Experimental study, *in* '2015 International conference on science in information technology (ICSITech)', IEEE, pp. 132–136.

Hew, K. F., Hu, X., Qiao, C. & Tang, Y. (2020), 'What predicts student satisfaction with moocs: A gradient boosting trees supervised machine learning and sentiment analysis approach', *Computers & Education* **145**, 103724.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Hutto, C. J. & Gilbert, E. (2014), Vader: A parsimonious rule-based model for sentiment analysis of social media text, *in* 'Eighth international AAAI conference on weblogs and social media'.

Klein, D. & Manning, C. D. (2003), Accurate unlexicalized parsing, *in* 'Proceedings of the 41st annual meeting of the association for computational linguistics', pp. 423–430.

Kobayashi, N., Inui, K., Matsumoto, Y., Tateishi, K. & Fukushima, T. (2004), Collecting evaluative expressions for opinion extraction, *in* 'International Conference on Natural Language Processing', Springer, pp. 596–605.

Krill, P. (2019), 'Python overtakes java on github'.
URL: *https://www.infoworld.com/article/3452666/python-overtakes-java-on-github.html*

Landolt, S., Wambsganss, T. & Söllner, M. (2021), A taxonomy for deep learning in natural language processing, Hawaii International Conference on System Sciences.

Le Glaz, A., Haralambous, Y., Kim-Dufor, D.-H., Lenca, P., Billot, R., Ryan, T. C., Marsh, J., Devylder, J., Walter, M., Berrouiguet, S. et al. (2021), 'Machine learning and natural language processing in mental health: Systematic review', *Journal of Medical Internet Research* **23**(5), e15708.

Li, X., Bing, L., Li, P. & Lam, W. (2019), A unified model for opinion target extraction and target sentiment prediction, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 33, pp. 6714–6721.

Liu, B. (2012), 'Sentiment analysis and opinion mining', *Synthesis Lectures on Human Language Technologies* **5**(1), 1–167.
URL: *http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016*

Liu, B. & Zhang, L. (2012), A survey of opinion mining and sentiment analysis, *in* 'Mining text data', Springer, pp. 415–463.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. & Potts, C. (2011), Learning word vectors for sentiment analysis, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies', Association for Computational Linguistics, Portland, Oregon, USA, pp. 142–150.
URL: *http://www.aclweb.org/anthology/P11-1015*

Mäntylä, M. V., Graziotin, D. & Kuutila, M. (2018), 'The evolution of sentiment analysis—a review of research topics, venues, and top cited papers', *Computer Science Review* **27**, 16–32.

Marcus, G. (2020), 'Gpt-3, bloviator: Openai's language generator has no idea what it's talking about'.
URL: *https://www.technologyreview.com/2020/08/22/1007539/gpt3-openai-language-generator-artificial-intelligence-ai-opinion/*

Maynard, D. & Greenwood, M. (2014), 'Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis'. The LREC 2014 Proceedings are licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.
URL: *https://eprints.whiterose.ac.uk/130763/*

McKinney, W. (2012), *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*, " O'Reilly Media, Inc.".

Meena, A. & Prabhakar, T. (2007), Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis, *in* 'European conference on information retrieval', Springer, pp. 573–580.

Mnasri, M. (2019), 'Recent advances in conversational nlp: Towards the standardization of chatbot building', *arXiv preprint arXiv:1903.09025* .

Montoyo, A., Martínez-Barco, P. & Balahur, A. (2012), 'Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments', *Decision Support Systems* **53**(4), 675–679.

Moore, A. & Rayson, P. (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', Association for Computational Linguistics, pp. 1132–1144.
URL: *http://aclweb.org/anthology/C18-1097*

Mäntylä, M. V., Graziotin, D. & Kuutila, M. (2018), 'The evolution of sentiment analysis—a review of research topics, venues, and top cited papers', *Computer Science Review* **27**, 16–32.
URL: *http://dx.doi.org/10.1016/j.cosrev.2017.10.002*

Naf'an, M. Z., Bimantara, A. A., Larasati, A., Risondang, E. M. & Nugraha, N. A. S. (2019), 'Sentiment analysis of cyberbullying on instagram user comments', *Journal of Data Science and Its Applications* **2**(1), 38–48.

Nazir, A., Rao, Y., Wu, L. & Sun, L. (2020), 'Issues and challenges of aspect-based sentiment analysis: A comprehensive survey', *IEEE Transactions on Affective Computing* pp. 1–1.

Nowak, J., Taspinar, A. & Scherer, R. (2017), Lstm recurrent neural networks for short text and sentiment classification, *in* 'International Conference on Artificial Intelligence and Soft Computing', Springer, pp. 553–562.

Pavlopoulos, I. (2014), 'Aspect based sentiment analysis', *Athens University of Economics and Business* .

Peng, H., Xu, L., Bing, L., Huang, F., Lu, W. & Si, L. (2020), 'Knowing what, how and why: A near complete solution for aspect-based sentiment analysis', *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(05), 8600–8607.
URL: *https://ojs.aaai.org/index.php/AAAI/article/view/6383*

Piryani, R., Madhavi, D. & Singh, V. K. (2017), 'Analytical mapping of opinion mining and sentiment analysis research during 2000–2015', *Information Processing & Management* **53**(1), 122–150.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M. & Eryiğit, G. (2016), SemEval-2016 Task 5: Aspect Based Sentiment Analysis, *in* 'International Workshop on Semantic Evaluation', San Diego, United States, pp. 19 – 30.
URL: *https://hal.archives-ouvertes.fr/hal-01838537*

Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M. & Benevenuto, F. (2016), 'Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods', *EPJ Data Science* **5**(1).
URL: *http://dx.doi.org/10.1140/epjds/s13688-016-0085-1*

Rietzler, A., Stabinger, S., Opitz, P. & Engl, S. (2019), 'Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification'.

Sak, H., Senior, A. & Beaufays, F. (2014), 'Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition', *arXiv preprint arXiv:1402.1128* .

Sanner, M. F. et al. (1999), 'Python: a programming language for software integration and development', *J Mol Graph Model* **17**(1), 57–61.

Schwartz, R., Dodge, J., Smith, N. A. & Etzioni, O. (2019), 'Green ai', *arXiv preprint arXiv:1907.10597* .

Soutner, D. & Müller, L. (2013), Application of lstm neural networks in language modelling, *in* 'International Conference on Text, Speech and Dialogue', Springer, pp. 105–112.

Strubell, E., Ganesh, A. & McCallum, A. (2019), 'Energy and policy considerations for deep learning in nlp', *arXiv preprint arXiv:1906.02243* .

Thet, T. T., Na, J.-C. & Khoo, C. S. (2010), 'Aspect-based sentiment analysis of movie reviews on discussion boards', *Journal of information science* **36**(6), 823–848.

Vaira, L., Bochicchio, M. A., Conte, M., Casaluci, F. M. & Melpignano, A. (2018), Mamabot: a system based on ml and nlp for supporting women and families during pregnancy, *in* 'Proceedings of the 22nd International Database Engineering & Applications Symposium', pp. 273–277.

Voutilainen, A. (2003), 'Part-of-speech tagging', *The Oxford handbook of computational linguistics* pp. 219–232.

Wiebe, J., Wilson, T. & Cardie, C. (2005), 'Annotating expressions of opinions and emotions in language', *Language resources and evaluation* **39**(2), 165–210.

Yaxiong, L., Jianqiang, Z., Deng, P. & Dan, H. (2014), 'A study of speech recognition based on rnn-rbm language model', *Journal of Computer Research and Development* **51**(9), 1936.

Yu, Q., Zhao, H. & Wang, Z. (2019), Attention-based bidirectional gated recurrent unit neural networks for sentiment analysis, *in* 'Proceedings of the 2nd International Conference on Artificial Intelligence and Pattern Recognition', AIPR '19, Association for Computing Machinery, New York, NY, USA, p. 116–119.
URL: *https://doi.org/10.1145/3357254.3357262*