Detecting Fake News by Leveraging Emotion

Johnny Scanlon

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Data Science)

Supervisor: Yvette Graham

Nov 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Johnny Scanlon

April 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Johnny Scanlon

April 19, 2022

Acknowledgments

I would first like to acknowledge the enormous support of my supervisor Yvette Graham for all her help throughout this project. Her advice and direction were incredibly helpful to this body of work.

I would like to thank my mother, Bridín, and my late father, Tim, without whom this would have been ten times the challenge. Their help and encouragement throughout college and beyond made all the difference. I am forever grateful for their assistance.

I would also like to acknowledge my friends Oscar and Harry for their continued support through the many challenges and hurdles encountered over the course of this project.

Johnny Scanlon

University of Dublin, Trinity College
Nov 2022

Abstract

Fake news has become an increasingly salient issue in the past number of years. The advent of the Internet and subsequent rise of social networks have provided a powerful platform for the rapid sharing of information. With this, the issue of widespread *misinformation* has become a serious problem.

Past work has shown that there is often a distinct emotional component to fake news. This project seeks to exploit this by explicitly leveraging information about the emotional content of news items in order to detect whether the news is "fake". The novel use of deepMoji is proposed as a means of detecting fake news. This architecture classifies emotions in text by returning probabilities of specific emojis being associated with the text. Emojis themselves are a powerful tool for conveying nuanced emotion across digital platforms. This project uses this to detect the differences in emotions conveyed between fake and real news.

Their use is examined in two specific ways. The first is to see how well these emotional features can be used to discriminate between real and fake news. The second is to see if they can be used to improve the results of popular state-of-the-art transformer models like BERT.

Through testing on the LIAR data set and a data set released as part of the AAAI Constraint-2021 shared task, it is found that these emotional features on their own are capable of discriminating between real and fake news. This confirms that there are distinct and measurable differences in the emotions typically conveyed in real and fake news. Positive results were found by combining this with BERT. However, it was not possible to confirm the statistical significance of this improvement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> "A liar begins with making falsehood appear like truth, and ends with making truth itself appear like falsehood."
>
> William Shenstone (1804)
> *"Essays on Men and Manners"*

1.1 An Overview of the Problem

"Fake news" has become a topic of much discussion and controversy in recent years. Its colloquial use as a term to describe news that contains "incorrect content or is of particularly low quality" led to it being awarded word of the year by Collins Dictionary in 2017.[1] Although fake news is not a particularly new issue, the rising discourse around it is a reflection of the increased impact this issue is beginning to have, particularly in socio-political spheres. To address this problem, it is important to understand the driving forces behind its rise and what makes it so harmful.

1.1.1 The Internet Age

The rise of the Internet has had a profound effect on the issue of fake news. Central to this is the massive effect of *social media* on the news ecosystem. In 2012, only 49% of

[1] `https://www.theguardian.com/books/2017/nov/02/fake-news-is-very-real-word-of-the-year-for-2017`

US adults reported seeing news on social media. By 2016, this figure was 62% and this trend appears to only be increasing.[2]. This trend is paired with a steady increase in the amount of hours the average person spends on social media. [3]

The gradual shift of the news media to the online sphere is not random. Social media's ability to rapidly share information makes it an ideal platform for news. In addition, it allows users to interact with news in a way that was not possible through newspapers and radio. This interactivity is in part what makes it such a powerful way to spread information. Information that people deem interesting or of note is shared and interacted with via comments. These new forces have dramatically altered the news ecosystem. This need for interaction in the form of shares and clicks pushes news agencies to create content that is more likely to be shared and clicked. This hunt for "virality" has led to a rise in things like clickbait.

Information that is shocking and quick to grab the reader's attention has been found to be more prone to go "viral". This can create an incentive for publishers to publish less nuanced and more attention-grabbing news.[4] In this environment, fake news can thrive. Misinformation, which can evoke an emotional response, is capable of spreading quickly before being disproven.

Consequently, the amount of misinformation shared on these social media platforms has increased dramatically. As an example, it was found that the most popular fake news was more popular on Facebook than the most popular real news during the 2016 US election.[5] Work by the Pew Research Centre has shown that the majority of social media users now expect the news they see on social media to be incorrect.[6]

Ultimately, the Internet and social networks have upended many long-standing modalities around news consumption. As this technology is almost certainly here to stay, the pressure is now on societies at large to adapt to the new pressures these technologies have introduced.

[2] http://www.journalism.org/2016/05/26/news-use-acrosssocial-media-platforms-2016/

[3] https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/

[4] https://www.bbc.com/news/uk-wales-34213693

[5] https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

[6] https://www.pewresearch.org/journalism/2018/09/10/news-use-across-social-media-platforms-2018/

1.1.2 Potential for Harm

Having discussed the societal changes that have led to an environment ripe for misinformation spread, it is important to discuss the impact of this. Highlighting *how* and *why* this issue is harmful helps to provide context for the academic interest in this area, as well as the work undertaken in this project. In addition, understanding how this problem becomes harmful can help explain the reasons for the approach taken in this project.

Regular exposure to misinformation has been found to affect the viewer's ability to discern real from fake news [1]. This, in turn, can have a damaging effect on users and society [2]. Having discussed the ubiquity of fake news in the last section, this paints a bleak picture.

The COVID-19 pandemic provides an excellent case study of this problem. The impact of misinformation was so great that the WHO (World Health Organisation) coined the term "infodemic" to describe it.[7] In their own words, infodemic describes "a situation where too much information including false or misleading information in digital and physical environments during a disease outbreak causes confusion and risk-taking behaviours that can harm health". They found that it also leads to mistrust in health authorities and undermines the public health response.

Elsewhere, events such as the Pizzagate scandal show the problem of fake news in the extreme. Here, the rapid spread of a piece of fake news led many to believe a conspiracy theory about high-ranking government officials in the run-up to the 2016 US presidential election. This tragically resulted in a shooting as a believer sought to uncover the conspiracy.[8] In this particular case, it is estimated that over a million tweets were made regarding "Pizzagate", showing how quickly such misinformation can spread.

Both of these examples show how this barrage of misinformation can damage a society and undermine expert advice. The extent to which this can impact a society is only now being fully understood. There are now real concerns about the effects of this fake news epidemic on the democratic process.[9] The revelation that state actors such as Russia have been contributing to this fake news epidemic to try and influence foreign elections has given grave cause for concern.[10] It seems that the malicious use of social media and fake news to destabilise social cohesion in targeted communities by state actors is no longer a fear, but a reality. Referring back to the opening quote by the English poet William

[7] `https://www.who.int/health-topics/infodemic`

[8] `https://www.rollingstone.com/feature/anatomy-of-a-fake-news-scandal-125877/`

[9] `https://www.theguardian.com/commentisfree/2021/feb/08/fake-news-facts-facebook-twitter-fox-news-democracy`

[10] `https://time.com/4783932/inside-russia-social-media-war-america/`

Shenstone, "A liar begins with making falsehood appear like truth, and ends with making truth itself appear like falsehood." Though published in 1802, it still eloquently captures the insidious nature of fake news in contemporary society. As fake news becomes more commonplace, it becomes much more difficult to tell fact from fiction in the news we consume. The profound impact this can have on social cohesion has spurred much of the academic interest in this area, as well as the work done in this project.

1.1.3 How is it being tackled?

This issue has not gone unnoticed by ruling bodies. Many of the social media giants have come under pressure to get a handle on the misinformation being spread through their platforms.[11] However, the manual detection of such fake news is far too labour intensive to ever be a real solution given the size of these sites.

Ultimately, effective tools for the automatic detection of fake news are required to get a handle on this issue. However, the detection of fake news on the Internet is not a trivial matter. This is made doubly complicated given that many instances of false news will be made with the intention of avoiding detection. While automatic detection has been rolled out to some degree by several major social networking sites, the continued prevalence of fake news shows more work is still needed.[12] Moreover, these methods still rely on a lot of manual checking of the content that is removed, which is highly resource-intensive. Several scandals around social networking sites have arisen recently concerned with the psychological suffering that content moderators face, with many reporting symptoms of PTSD as a result of spending day after day viewing this content. The need to limit the deleterious impacts faced by human moderators in the process provides more motivation for the development of improved automatic moderation.[13]

1.2 Project Goal and Motivation

The goal of this project is to develop an effective deep learning approach to automatic detection of fake news as would be encountered in an online setting by explicitly leveraging emotion.

[11] `https://www.theguardian.com/technology/2018/nov/27/facebook-fake-news-inquiry-the-countries-demanding-answers`

[12] `https://about.fb.com/news/2021/03/how-were-tackling-misinformation-across-our-apps/`

[13] `https://www.bbc.com/news/technology-52642633`

This project proposes the novel use of the deepMoji [3] emotion detection system as an aid to fake news detection. The project examines how this model can perform on its own. In addition, it examines whether it can be used to improve results for established transformer methods.

By looking at how deepMoji performs on its own, I seek to estimate the extent to which emotional differences can be used to differentiate fake from true news. If successful, this may also reveal something as to the kinds of emotions and sentiments that are useful for differentiating fake news. In addition, by combining this with BERT, it can be seen whether this emotional information can be used to help improve state-of-the-art results.

Through training on over a billion tweets, deepMoji predicts emotion by returning the probability of specific emojis being associated with some input text. Emojis themselves are a product of the digital age, whose explicit use is to convey the emotion of a speaker. This was a key motivating factor for its use in this system, as this would hopefully make deepMoji particularly attuned to the detection of emotion from text across digital platforms.

The motivation for leveraging emotion as a means of detection is to try to address some of the difficulties in fake news detection; namely, that fake news can be nuanced and may be intentionally trying to avoid detection. Details about fake news and its characteristics are discussed in more detail in Chapter 2.

The central thesis of this project is that by focusing on an underlying common element of fake news, such as emotion, its detection can be improved. If fake news tries to get us to think with our hearts and not our heads, then this project seeks to exploit that.

Chapter 2

Literature Review

2.1 Fake News and Social Media

Having given an overview of the problem in Chapter 1, this section seeks to give a more in-depth review of fake news and how it exists on social media.

Social media has become an increasingly popular source of news. From 2012 to 2016, the percentage of adults in the US who reported consuming news on social media increased from 49% to 62%. By 2018, this figure was found to be up to 68%.[1] This is paired with a steady increase in the amount of time people spend using social media.[2] Social media has now become a major source of news for many, and this trend seems set only to increase. The reasons for social media's use as a mode of sharing news are two-fold [4]:

- It is generally quicker and less expensive to consume news on social media compared to traditional news media, such as television or newspapers.
- User engagement is easier. Social media allows users to share, comment on, and discuss news with others very easily.

It is for these same reasons that social media has become a potent source of misinformation. This fake news can have a damaging effect on individuals and society [2]. Fake news can cause people to be misled and accept false beliefs [5]. It can also affect the way consumers react to true news by impacting viewers ability to assess the veracity of

[1] https://www.pewresearch.org/journalism/2018/09/10/news-use-across-social-media-platforms-2018/

[2] https://www.socialmediatoday.com/marketing/how-much-time-do-people-spend-social-media-infographic

news pieces [1]. Consequently, it can damage the entire news ecosystem, lowering the trustworthiness of many news sources. The social context and how social media plays a role in this is explored in depth in [1]. This paper describes a *tri-relationship* between publishers, news pieces, and users, with each influencing the other.

This rising tide of misinformation has been accompanied by an increased academic interest in the area of fake news detection. However, before any work can be done in this area, it is necessary to clearly specify what is meant by the term "fake news". Although this term has seen widespread use in recent years, it lacks a rigorous academic definition. Accordingly, many definitions have been used. Often, these definitions disagree on whether to include things like satire and hoaxes in their definition of fake news.

This project uses a narrower definition of fake news, defined as news that is intentionally and verifiable false [6]. This definition has seen use in a number of studies [4]. This definition makes *authenticity* and *intent* a requirement for something to be labelled as fake news. It explicitly does not include topics like satire.

Looser definitions have also been used where any false information qualifies a piece for being fake news [7][8]. While such definitions may be useful in an academic context, this is too broad to be used in the context of this project. The aim of this project is to develop a model which would be be useful in the identification of fake news in a real setting. For example, satire, which intentionally reveals its bias, often for comedic effect, would not fall under what most consider to be fake news. A model that labels such pieces as fake news would not have much practical use in a live setting.

2.2 Datasets

As with many machine learning problems, the task of fake news detection is dependent upon the existence of available training data. For supervised approaches, this requires laboriously annotated datasets. Although several of these datasets have been put together with the goal of fake news detection in mind, they differ over several key factors. In particular, the content and the number of truth grades vary considerably. "Content" here refers to the type of data that is being evaluated for veracity. This content can range from short tweets to entire articles. "Truth grades" refer to how many categories of "truthfulness" are assigned. This can range from binary true or false to more nuanced "mostly true" or "mixture of true and false" labels. This section seeks to provide a brief overview of some popular datasets and their respective details. Finally, it specifies the datasets used in this project and provides a justification for their selection.

- **LIAR** [9]: This dataset provides around 12,000 short statements accompanied by a finely-grained truth rating. In addition to these statements, metadata, such as source, context, and justification, is also included.

- **AAAI Contraint@2021** [10]: This data set was released as part of the AAAI Contraint@2021 open challenge. This challenge examined the fake news surrounding the Covid-19 crisis. This data set contains pieces of text concerning Covid-19 with an accompanying binary truth label.

- **CREDBANK** [11]: This is a large database of approximately 60 million tweets. These tweets were made over a period of about three months, beginning around October 2015. These tweets are related to 1049 events and are each given a truth rating by 30 annotators.

- **BuzzFeedNews**:[3]This data set contains details about several thousand articles published over several weeks in the lead up to the 2016 US presidential election. The dataset itself does not contain any article or comment content itself, but rather details around the number of shares and sources of the article. This makes it unsuitable from a text classification task perspective, as in the case of this project.

- **BuzzFace** [12]: This is an extension of the previous BuzzFeedNews article that contains more metadata, including the article and the comment section content. This makes it more suitable from a text classification perspective.

[2] goes into greater detail on the individual contents of these datasets. For this project, the LIAR and AAAI datasets provide the required level of content and are of reasonable size. Using several data sets with different truth grades also has the benefit of testing the ability of the model to a greater extent. Naturally, simple binary truth grades are easier to classify than more nuanced ones.

Larger datasets, such as CREDBANK, were not considered due to computational constraints. Data sets such as BuzzFeedNews were not considered because they did not contain any article text. This makes them unsuitable for the approach taken in this project. Data sets containing comment sections were also not of use. The approach taken in this project aims to focus on using cues in the articles themselves to make judgments on their veracity.

Ultimately, it was deemed that the LIAR and AAAI data sets best suited the aims of this project.

[3]https://github.com/BuzzFeedNews/2016-10-facebook-fact-check/tree/master/data

It is worth considering how bias can be introduced into these datasets. Bias can be introduced into a data set through a number of factors, including selection bias and decisions around truth labels. Controlling for these factors has been shown to reduce the performance of fake news detection models [13]. The decision to use several data sets hopes to counteract any possible bias inherent in these data sets. Possible bias within these datasets is treated in more depth in Chapter 3.

2.3 Core Neural Network Concepts

The aim of this section is to briefly explain some of the core ideas behind neural networks, as well as some of the concepts that are referenced throughout this project. The aim is not to provide an in-depth description of these concepts, but merely to make the reader familiar with them so that the rest of the paper may be properly understood.

2.3.1 Neural Networks

Inspired by neural connections in the brain, artificial neural networks are a subset of machine learning algorithms. Although many variations exist, they are typically composed of layers of nodes. The connections to these nodes each have associated weights, and the node itself has a threshold that must be reached before it can fire. By weighting these connections, the node can give weight to the aspects of the input data that are important to the problem it is trying to solve and reduce the weight to the aspects of the input data that are not as significant [14]. The non-linearity of this firing mechanism allows neural networks to model complex problems.

By combining many of these nodes into layers, deep neural networks are formed. An example of this is shown in Figure 2.1 [15]. These deep neural networks are capable of modelling complex problems and have had great success in solving many AI (Artificial Intelligence) problems.

2.3.2 Training

The training of neural networks involves the learning of weights that allow it to solve the problem at hand. In order to train such machine learning models, it is typically necessary to calculate the gradient with respect to each weight so that the algorithm may change the weights in order to reduce some loss function. Given that neural networks can easily

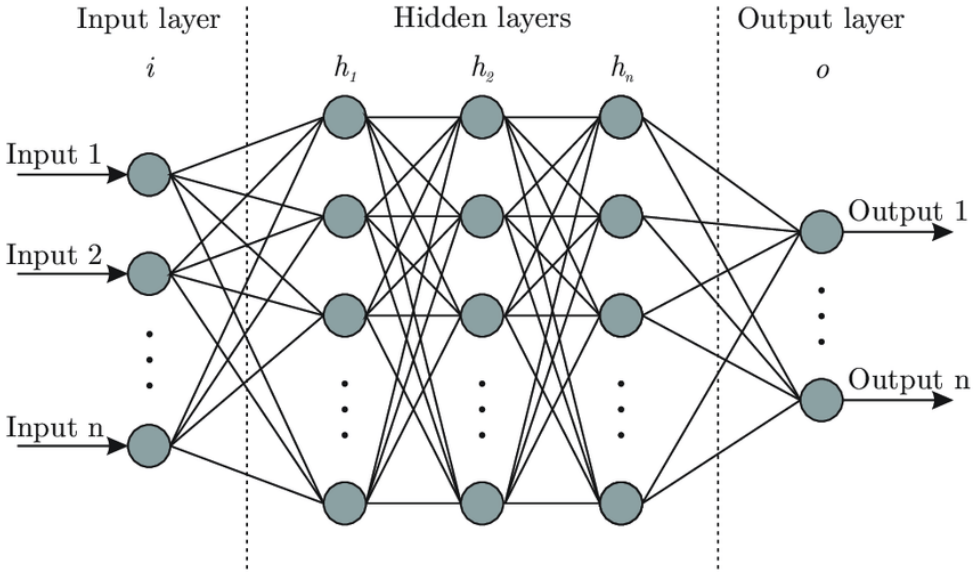


Figure 2.1: A diagram of a deep neural network with hidden layers [15]

contain a great number of weights, this task is nontrivial. This task is solved by the backpropogation algorithm [16]. Backpropogation allows for the efficient computation of these gradients through the use of the chain rule. This allows the algorithm to compute gradients a single layer at a time and remove a huge amount of redundant calculations. However, this can have its own issues, where the computations of gradients are reliant on the gradients that have come before them. This means that issues in the computation of some gradients can easily propagate backward and affect many of the weights behind them.

2.3.3 Transfer Learning

Transfer learning [17] refers to the use of information learnt in one task, being applied to another similar task. For example, a model trained to detect dogs in photos may be used for the similar task of detecting cats with relatively little retraining. Typically, transfer learning refers to large models trained on large amounts of data for a more general task being applied to more specific tasks. The ability of transfer learning to take knowledge from these general tasks and apply it to specific instances is of huge benefit to the machine learning community. Take, for example, the issue of fake news detection that is examined in this project. It is infeasible to have a fake news corpus with hundreds of millions or even billions of reliably labelled fake news. As such, being able to use models that have been trained on corpora of such sizes and have managed to identify useful features in language is of immense use to this problem.

2.4 Fake News Detection

This section seeks to provide an overview of past attempts at detecting fake news using NLP (natural language processing) techniques. In addition, it outlines the current state-of-the-art while contrasting the approach taken in this project.

2.4.1 Early Techniques for Fake News detection

Analysis of straightforward linguistic features has revealed differences between fake and real news [18]. For example, fake news was found to be more likely to use first-person pronouns such as "I" and "we". It was found to contain exaggerative language, using more subjectives, superlatives, and modal adverbs than real news. Modelling with these features was found to perform better than simple random baseline models.

These straightforward approaches helped inspire the approach taken in this project, which aims to pair more complex NLP approaches with methods that specifically target characteristics known to be associated with fake news. More specifically, this project aims to exploit previously observed differences in emotional content between fake and real news. The work done in [18] showed that emotion can be a strong indicator of fake news.

2.4.2 Word Embeddings

Moving beyond these simple linguistic features, the development of advanced word embeddings [19] has been very useful for the task of fake news detection. Word embeddings are ways to tokenise and encode words in a way that the word and its meaning can be more easily understood by a computer. This is typically done by converting the word token into a *word vector*. This moves away from simple tokenisation and tries to build some form of feature space in which word meaning can be encoded. Some significant approaches to this are outlined below.

- **One Hot Encoding**: Given a dictionary of N unique words, this method simply uses a vector of length N, with each bit corresponding to a particular word. When a given bit is 1, the vector represents that word. This does not scale particularly well as each word must be represented by a very long and very sparse vector.
- **TF-IDF**: TF-IDF (term frequency - inverse document frequency) looks at using a word's frequency in a given document, compared to its frequency in all documents,

to try and draw conclusions.

For example, if a model was trying to predict positive or negative sentiment in reviews, it might notice that the word "love" tended to occur once every 10 words in positive reviews and once every 500 words in all reviews. From this, it could be surmised that the word love is likely to indicate a positive sentiment. Language is, of course, more complicated than this. Think of the phrases "I hate that I love it" and "I love to hate it". These might look very similar to a machine, but their sentiment is actually quite different. Despite this, TF-IDF is still widely used in many NLP tasks and performs quite well [20].

- **word2vec and GloVe**: These are short word vectors whose values are determined through deep learning methods. These have proven to be very effective ways of representing word meaning in a manner that computers can interpret. In particular, these received a lot of attention for their use in word similarity and semantic relationship tasks. An example of this is shown in Figure 2.2 [21]. Further improvement of these embeddings has been shown to improve the results of fake news detection and other NLP problems [22].

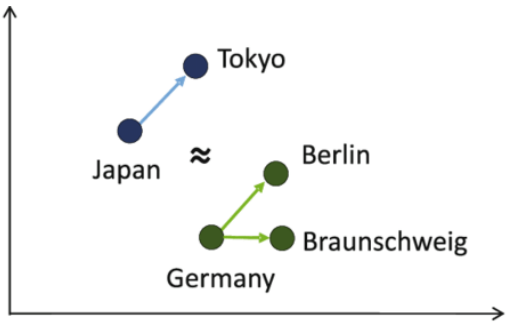


Figure 2.2: An example of how word vectors capture meaning and can be used in word similarity and semantic relationship tasks [21].

Combining these word vectors with classification models proved to be very effective for many text classification problems, including fake news detection. Machine learning techniques such as SVM (Support Vector Machines) and deep learning proved to be very effective as classification models on top of these word vectors.

Although these word embeddings proved to be very effective in capturing the individual meanings of words, these methods fail to meaningfully capture the sequential nature of language.

2.4.3 Recurrent Neural Networks

RNNs (Recurrent Neural Networks) are a type of neural network specifically designed to handle sequential data [23]. This made them ideal for language processing, and they quickly set the state-of-the-art for a host of natural language processing tasks. They use an internal memory state, so that in theory they can learn details for arbitrarily long sequences. A high-level view of this model is shown in Figure 2.3 [24]. The output of the internal memory state, denoted by h in the figure, is fed back when calculating its new state. The model can be unfolded into a classic feedforward neural network. In figure 2.3, this unfolding into a feedforward network is clearly shown.

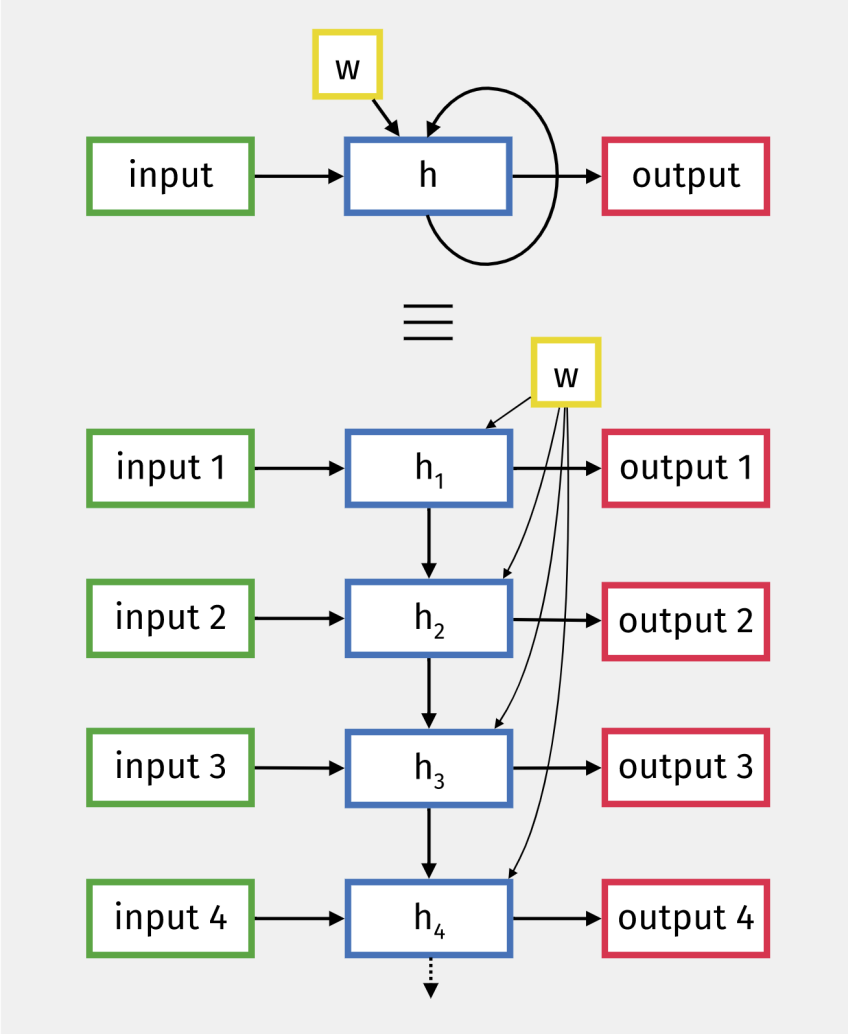


Figure 2.3: A high Level View of an RNN Model [24].

Although this works well in theory, basic RNNs suffer from several issues. In particular, the problem of vanishing and exploding gradients proved to be a significant issue. This problem is encountered when using backpropagation to learn model weights. As networks

became longer/deeper, gradients tend to either explode or go to zero. This can affect the learning of any weights behind that weight in the network [25]. This prevented the implementation of RNNs that were too long. In addition, the structure of RNNs makes parallelisation very difficult. This made building large models prohibitively expensive computationally. Furthermore, RNNs are particularly unsuitable for *transfer learning*.

Some of these issues were overcome in part through the development of **LSTM** [26] and **GRU** [27]. These models use feedback connections to partially solve the vanishing gradient issue. While this saw a dramatic increase in performance, these models still suffer from some of the same problems as RNNs. Namely, exploding gradient and difficulty with transfer learning.

Traditionally, RNN models based on LSTM and GRU have proven to be very effective in the realm of fake news detection. Examples of effective LSTM and GRU models are shown in [18] for the LIAR data set.

2.4.4 Transformers

Since the advent of transformers in 2017 [28], many of the leading approaches have used this technology. Transformers address several key issues of these classic RNN approaches. They have a constant computational cost to learn distant dependencies. In addition, their architecture makes them easy to parallelise. On top of this, they are especially suited to transfer learning. This allows pre-trained models, trained on huge corpora, to be reused for other more specific tasks.

Transformers are based on the mechanism of self-attention. This mechanism weights parts of the incoming sequence with other parts of the sequence. The original idea of attention was to create a mechanism that weighted the incoming data on the basis of its impact on the problem at hand. In this way, it mimicked the process of cognitive attention, after which it is named. This process had been used to assist other methods [29] before it was shown that attention alone was a powerful tool in [28].

In practise, this self-attention is implemented using scaled dot-product attention. This uses three weight matrices: query weights W_Q, key weights W_K, and value weights W_V. For each word token i, the word embedding w_i is multiplied with each of these matrices to generate three vectors: a query vector $q_i = x_i W_Q$, a key vector $k_i = x_i W_K$, and a value vector $v_i = x_i W_V$. To get the attention weights between a token i and a token j, the dot product of q_i and k_j is found. In order to stabilise the gradients for training, these attention weights are divided by the square root of dimension of the key vector. These

attention weights are then normalised using the softmax function.

Having separate query and key weight matrices is significant, as it allows for asymmetric relationships. That is, the dot products $q_i \cdot k_j$ and $q_j \cdot k_i$ may not be the same. This means that although a word token i may attend strongly to token j, this does not mean that token j attends strongly to token i.

These matrix operations can be expressed through the following equation:

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

Although a single attention computation can be useful in seeing how strongly words relate to each other, the true power of transformers comes from the use of *multi-attention heads*. This architecture uses several attention modules in parallel with each other. In practise, this allows different attention heads to attend to different aspects of language. For example, one attention head may look for verb-object relationships, while another may look for adjective-noun relationships. A diagram showing a single attention computation model as well as a multi-head attention model is shown in Figure 2.4 [28].

To highlight how these multi-attention heads attend to different linguistic aspects for a given word, the results of a multi-headed attention unit are shown in 2.5 [28].

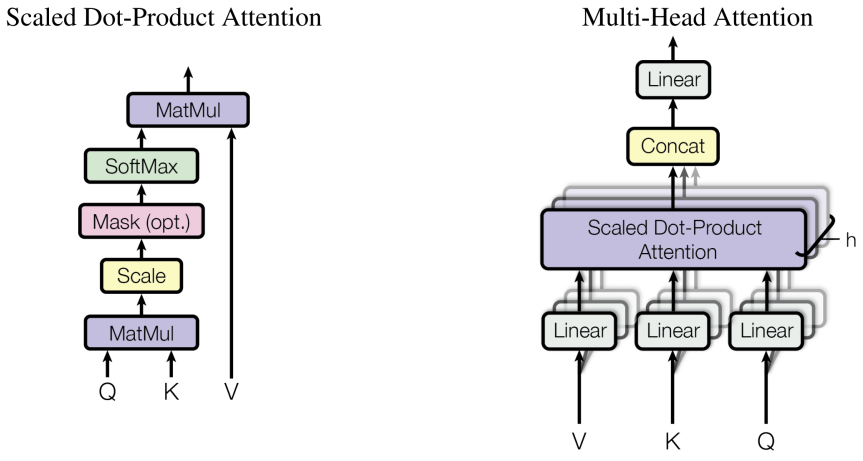


Figure 2.4: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel [28].

These multi-head attention units are very easy to parallelise. This makes training larger models on huge amounts of data much more feasible than with other RNN based models. In the years that followed, many large-scale pre-trained models were made available. These models were trained on huge amounts of data, and so had developed weights

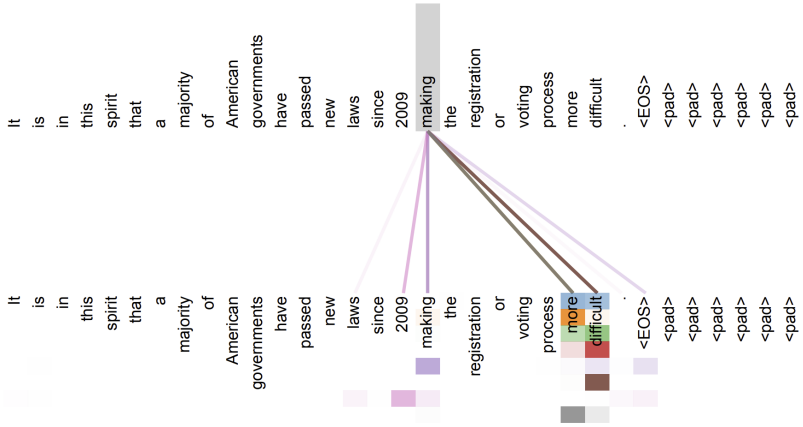


Figure 2.5: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colours represent different heads. Best viewed in colour [28]

capable of detecting very useful linguistic features. For example, BERT [30], a highly popular model, was trained using BooksCorpus (800M words) [31] and English Wikipedia (2,500M words). By fine-tuning these huge pre-trained models for specific problems, immense success in a wide range of NLP tasks was found.

Evidence of their success can be seen by looking at the result of the AAAI Constrain-2021 shared task for fake news detection [32]. The model that took the first place used a transformer model based on Covid-Twitter-Bert [33]. Covid-Twitter-Bert is a variation of BERT trained with additional tweets about the Covid-19 epidemic [34]. This is an excellent example of the power of transfer learning and transformers. The model that took third place used an ensemble method of many pre-trained transformer models, including Bert, roBerta, and Ernie. [35]. This paper also proposes a bidirectional LSTM model. The ensemble transformer model proved to be the more effective model. This highlights the effectiveness of transformers in text classification.

2.4.5 Alternative Approaches

It is also worth considering alternative methods for fake news detection. These may go beyond the straightforward classification of texts. For example, user interactions have been thought of as a network to be unfolded into deep learning models [36]. Such methods make use of the "social" aspect of social media, using user behaviour as an indicator of

fake news.

2.5 Emotion Detection

Emotion detection from text has proven to be a significant task in the field of NLP. As a kind of sentiment analysis, it also saw improvement through the development of RNN models and later through transformer models [37].

2.5.1 Emotion Detection for detecting Fake News

Regarding the detection of fake news by leveraging emotional features, some previous attempts have been made. Two significant approaches, in particular, are highlighted here.

[38] uses an LSTM model to detect emotional features. To build this emotion detection, the model is trained on a data set built from New York Times articles and their corresponding Facebook reactions. This goes against the underlying motivation of this project, which is to try to detect an underlying publisher emotion. Using Facebook reactions, this model is really being trained on reactions to the news rather than the emotions inherent in the news article itself. While this may prove to be an effective method for Fake News detection, this may fall into the trap of modelling user outrage. As effective as it may be to model user outrage, it is important to note that true news is not always popular and fake news is not always unpopular. This project explicitly seeks to avoid modelling for emotions a text may elicit and instead focusses on the emotions conveyed in the text itself.

[39] models both the published and viewer emotions. Using an LSTM approach, this model creates a *dual-emotion* feature vector of both publisher and viewer emotions, which is used to bolster the decisions of another model. This is similar in nature to the approach taken in this project. This is useful as it allows the effect of emotion on other models to be easily identified. While this is a better approach in that it more explicitly analyses the targeted emotions, it still models for user reaction by analysing comment sections. Again, this is something this project aims to avoid. Modelling user comments brings with it a greater risk of modelling user class, background, and affiliation. Hence, this project aims to limit its detection to merely the news being discussed.

In both cases, combining the resulting emotional features with other text classification models managed to improve the results. Seeing the success of these approaches influenced

the decision to help develop meaningful emotional features using emojis in this project.

Ultimately, the role of emotion as a means of fake news detection seems promising, especially when used in conjunction with other NLP techniques. This project aims to use transformer methods for fake news classification along with emotion detection to see if emotion detection can be used to improve results.

2.5.2 Pre-Trained Emotion Detection Models

This project makes use of pre-trained emotion detection models. Similarly to the pre-trained language models discussed in the previous subsection, these are models specifically trained to detect emotion.

Referring back to the models discussed in the previous section, [39] uses an emotion model put forward by NVIDIA.[4] This model models for eight primary emotions.

Similarly, the EmoCred emotion detection model presented in [38] models for 5 emotions: love, joy, surprise, sadness, and anger.

It was felt that, in reality, this was quite a narrow view of human emotions. This inspired the decision to use DeepMoji as an emotion detection model.

DeepMoji [3] is a model trained on 1.2 billion tweets that associates emojis with text. In digital environments, emojis are used to convey emotion. This partly inspired their use in this project, as they may be particularly suited to detecting the emotion in fake news which is spread through these digital environments. An example of how deepMoji associates emojis with words can be seen in Figure 2.6. It has achieved state-of-the-art performance in many nuanced emotion detecting tasks, such as sarcasm detection. To the best of my knowledge, this is the first application of deepMoji in the realm of fake news detection.

[4] `https://github.com/NVIDIA/sentiment-discovery`

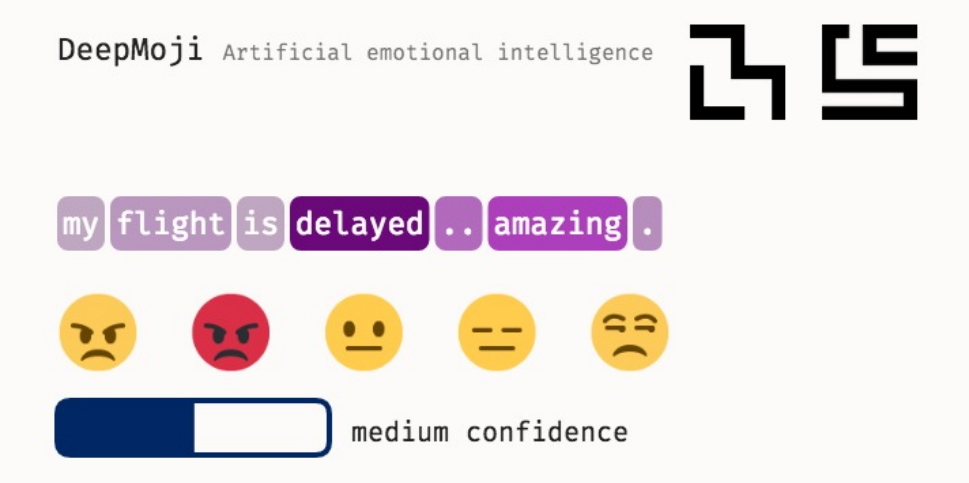


Figure 2.6: Example output of the deepMoji model

Chapter 3

Datasets

This project uses the LIAR and AAAI datasets. While a brief overview of these datasets was provided in Chapter 2 as part of the literature review, this section seeks to provide a more in-depth analysis of these datasets. As well, it discusses possible sources of implicit bias within these datasets that may affect the detection of fake news.

3.1 Dataset Analysis

3.1.1 AAAI Analysis

The AAAI was released as part of the shared task at Constraint-2021. This data set contains 10,700 manually annotated social media posts and articles. They are each given a binary "real" or "fake" label. Some examples of these text items and labels are shown in Table 3.1.

There are slightly more occurrences of real data points than fake data points in all test sets. Although it is only a slight bias, it should be considered when evaluating the results. The exact numbers of real and fake data points across training, validation, and test sets can be seen in Table 3.2.

TF-IDF (Term Frequency - Inverse Document Frequency) analysis was performed for both real and fake labels in the training set. The aim here is to get a feel for the linguistic disparities between the two classes. The results are shown in word clouds in Figure 3.1.

From these word clouds, it is possible to see that there is a distinct difference in the nature of language used in real and fake data points. For "real" news, geographical place names are particularly prevalent. For example, "Lagos" stands out dramatically. Looking

Label	**Source**	**Text**
Fake	Facebook	All hotels, restaurants, pubs etc will be closed till 15th Oct 2020 as per tourism minister of India.
Fake	Twitter	#Watch Italian Billionaire commits suicide by throwing himself from 20th Floor of his tower after his entire family was wiped out by #Coronavirus #Suicide has never been the way, maysoul rest in peace May God deliver us
Fake	Fact Checking	Scene from TV series viral as dead doctors in Italy due to COVID-19
Fake	Twitter	It's being reported that NC DHHS is telling hospitals that if they decide to do electivesurgeries, they won't be eligible to receive PPE from the state. The heavy hand of government. I hope Secretary Cohen will reverse course. #NCDHHS #COVID19NC #ncpol
Real	Twitter (WHO)	Almost 200 vaccines for #COVID19 are currently in clinical and pre-clinical testing. Thehistory of vaccine development tells us that some will fail and some will succeed-@DrTedros #UNGA #UN75
Real	Twitter (CDC)	Heart conditions like myocarditis are associated with some cases of #COVID19. Severecardiac damage is rare but has occurred even in young healthy people. CDC is working-tounderstand how COVID-19 affects the heart and other organs.
Real	Twitter (ICMR)	ICMR has approved 1000 #COVID19 testing labs all across India. There was only one government lab at the beginning of the year. #IndiaFightsCorona. #ICMRFightsCovid19

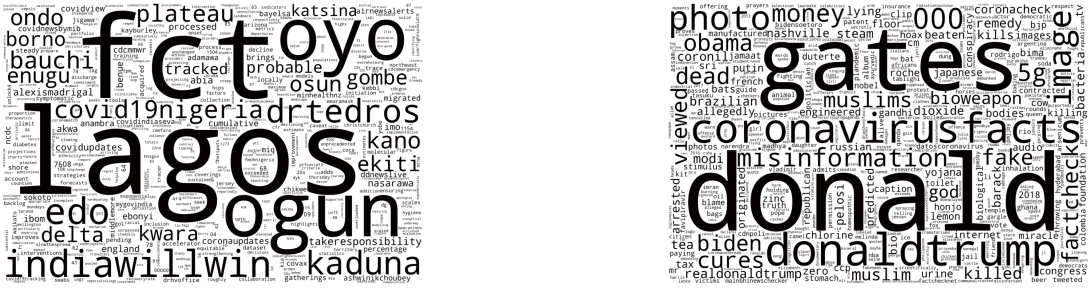
Table 3.1: Examples of Real and Fake datapoints from the AAAI dataset

deeper, more place names like England, Christchurch, and Borno also appear. In addition to this, scientific words related to the Covid-19 response appear. For example, words such as "plateau" and "immunization".

In contrast to this, fake news items show a particular proclivity for political and celebrity figures. For example, "Trump" is the most significant word. "Obama" and "Gates" (Bill Gates) are also popular words. That such political figures are commonly used in conjunction with misinformation is not surprising. Especially if we consider that often misinformation is frequently used as a means to push a particular agenda. Perhaps what is surprising is that words such as "misinformation" and "fake" are so closely tied to fake news itself. Delving deeper, this can begin to make sense when considering that fake news will often go against establishment facts. In doing so, it may be quick to label other points of view as fake in an effort to discredit them. Ironically, it seems that the

set	Real	Fake	Total
Training	3360	3060	6420
Validation	1120	1020	2140
Test	1120	1120	2040
Total	5600	5100	10,700

Table 3.2: Label distribution across all sets of AAAI data



(a) Real (b) Fake

Figure 3.1: TF-IDF word clouds for real (a) and fake (b) labels in the AAAI training set

presence of the hashtag "#coronavirusfacts is a great indicator that a particular tweet contains fake news. In addition, dramatic language such as "kills" and "bioweapon" is also prevalent. An indicator of the exaggerative language commonly used in Fake News, as discussed in Chapter 2. Following these observations, it is perhaps unsurprising that terms such as "Muslim", "Russian", and "CCP" (Chinese Communist Part) feature strongly. If the frequent aim of misinformation is to direct panic to push an agenda, it is perhaps unsurprising that such hot-button topics appear frequently.

3.1.2 LIAR Analysis

The Liar dataset was released in 2017 and contains 12,800 manually labelled short statements. These short statements are taken from `politifact.com`. [1] This is a Pulitzer prize-winning website that provides detailed judgments with fine-grained labels for claims. Unlike the AAAI dataset, these statements concern many different topic areas and contexts. The LIAR data set contains metadata surrounding the topic, such as speaker and occasion. As well, it contains a detailed explanation of what is wrong with the statement. As explained in Chapter 2, this metadata is ignored in favour of a text based approach as would be encountered in the wild. To get a better sense of the data contained within

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

the dataset, several excerpts are shown in Table 3.3. In this instance, the metadata is included so that the contents of the dataset can be better understood. However, much of this information is ignored for the approach taken in this project.

Statement: *"The last quarter, it was just announced, our gross domestic product was below zero. Who ever heard of this? Its never below zero."* **Speaker**: Donald Trump **Context**: presidential announcement speech **Label**: Pants on Fire **Justification**: According to Bureau of Economic Analysis and National Bureau of Economic Research, the growth in the gross domestic product has been below zero 42 times over 68 years. Thats a lot more than "never." We rate his claim Pants on Fire!
Statement: *"Newly Elected Republican Senators Sign Pledge to Eliminate Food Stamp Program in 2015."* **Speaker**: Facebook posts **Context**: social media posting **Label**: Pants on Fire **Justification**: More than 115,000 social media users passed along a story headlined, "Newly Elected Republican Senators Sign Pledge to Eliminate Food Stamp Program in 2015." But they failed to do due diligence and were snookered, since the story came from a publication that bills itself (quietly) as a "satirical, parody website." We rate the claim Pants on Fire.
Statement: *"Under the health care law, everybody will have lower rates, better quality care and better access."* **Speaker**: Nancy Pelosi **Context**: on 'Meet the Press' **Label**: False **Justification**: Even the study that Pelosi's staff cited as the source of that statement suggested that some people would pay more for health insurance. Analysis at the state level found the same thing. The general understanding of the word "everybody" is every person. The predictions dont back that up. We rule this statement False.

Table 3.3: Example data entries for LIAR dataset

As seen from the examples in Table 3.3, the LIAR data set uses a more fine-grained form of truth grading than the AAAI data set. In total, six distinct truth grades are specified. These are listed in order of most fake to most true below.

- pants-fire
- false
- barely-true
- half-true

- mostly-true
- true

The LIAR data set is relatively well balanced. With the exception of the pants-fire class, which has only 1,047 instances, instances for all other classes range from 2,053 to 2,627. Precise statistics can be seen in Table 3.4.

set	**pants-fire**	**false**	**barely-true**	**half-true**	**mostly-true**	**true**	**Total**
Training	839	1995	1654	2114	1962	1676	10,240
Validation	116	263	237	248	251	169	1284
Test	92	249	212	265	241	208	1267
Total	1,047	2507	2103	2627	2454	2053	12,791

Table 3.4: Label distribution across all sets of LIAR data

There are two possible approaches for this data set. One is to leave the classes as they are and test the model for its ability to detect finely grained truth classes. The other is to pool classes together. In this was the dataset can be converted into a simpler detection problem. If the classes are pooled into simple true and false classes, this becomes a straightforward binary classification task.

In order to create word clouds visualising significant words in the underlying data, this pooling method is used to convert the dataset into binary true and false classes. This poses the problem of which classes to include in true and false. For these word clouds, "pants-fire", "false", and "barely-true" are pooled into a false class, and "half-true", "mostly-true" and "true" are pooled into a true class. This highlights an underlying issue in fake news detection: where to draw the line in the sand between what is false and what is true. It is unrealistic and impractical to say that all information posted on a social media site must be rigorously true. Conceptualising truth as a sliding scale means that in some sense what is true and what is not is left up to the discretion of the person building the model. For the sake of these word-clouds the line was drawn at "barely-true". This has the benefit of making the datasets more balanced. However, analysis shows that the entries under the "barely-true" label were beyond what I would be comfortable calling true for the sake of this project.

An example of a barely true statement is given here: *"Says Vice President Joe Biden 'admits that the American people are being scammed' with the economic stimulus package."* This statement came from John Boehner, a House Republican Minority Leader. Politifact found this to be a reference to Biden discussing the issue of people being falsely promised

stimulus grants in return for personal information by con artists.[2] This is a very different sentiment to John Boehners statement, which seems to imply that the economic stimulus plan is a scam in and of itself. This kind of manipulation of words to push an agenda is typical of fake news and hence why statements from the "barely-true" class were classed as fake.

The resulting word clouds made from the training set can be seen in Figure 3.2.

(a) Real (b) Fake

Figure 3.2: TF-IDF wordclouds for real (a) and fake (b) labels in the AAAI training set

Compared to the word clouds for the AAAI dataset, many of these words seem much more general. This makes sense when considering that the AAAI data set focusses solely on a single topic (COVID-19), while the LIAR data set covers a range of topics and contexts. Looking deeper, however, some similar trends emerge. The fake news word cloud highlights words like "Jews", "Socialists", and "Corruption. In a way this mimics the use of "Muslims", "CCP", and "misinformation" that were so prevalent in the AAAI dataset fake news word cloud. From both datasets, it seems that fake news has a proclivity for certain topics. Looking deeper reveals the use of words like "Genocide". Again, showing the use of extreme language. Specific people are frequently mentioned. For example, "Soros" is frequently mentioned, referencing the controversial billionaire investor George Soros.

Although used for analysis here, this pooling of truth labels is avoided when implementing the models proposed in this project. The AAAI dataset already provides a basis for assessing how the models perform in the binary case. The finally grained truth labels of the LIAR dataset are useful for determining how well the models can perform for more nuanced truth labels.

[2] `https://www.politifact.com/factchecks/2009/jun/08/john-boehner/boehner-claims-biden-said-stimulus-scamming-americ/`

3.2 Dataset Bias

It is important to consider possible bias within the dataset. These are any underlying biases that could make detection easier. For example, if all the fake news items in a given data set are from a particular political party, the model may just end up associating talking points from that party with fake news. The model may perform very well on the data set but, in practise, work poorly. There is also an insidious element to this. If there is a bias in the data set towards certain political affiliations, this could lead to the model unfairly silencing points of view by being more inclined to label them as fake news. Careful thought was given to the datasets chosen for this very reason. Both the AAAI and LIAR dataset are published by reputable sources and have seen widespread use across the field. Additionally, using two datasets helps to check against any possible bias. Regardless of there widespread adoption, this subsection seeks to investigate obvious sources of potential bias. While it is impossible to definitively say that a data set is without bias, the aim of this subsection is to ensure that a certain standard was reached in terms of obvious bias.

3.2.1 Investigating Bias for AAAI

Given that the AAAI data set contains only text statements and labels without any metadata, it is particularly difficult to assess this data set for potential bias.

The Real News in the AAAI dataset are tweets from verified sources and give useful information on COVID-19. The fake news items are tweets, posts, articles which make claims and speculations about COVID-19 which are verified to be not true. This fake news was verified as false through popular fact verification websites like PolitiFact and Snopes.[3] As such, we can reasonably be confident that the individual data points are correctly labelled.

One possible source of Bias, which could make it far easier for models to differentiate between real and fake news, is the nature of these sources. If true news is only taken from verified accounts, and fake news from unofficial accounts, what a model may in reality be learning is the difference in language between these two. Although this is difficult to mitigate against, the use of appropriate baseline models can help determine how well a model is performing with more confidence.

Although it is difficult to trawl through the actual dataset and confirm that it is balanced, a point that stood out from the word clouds in Figure 3.1 was that geographic place names were strongly correlated with true news. While there is nothing inherently

[3] `https://www.snopes.com/`

wrong in this, and it could be an indication that real Covid-19 posts were more likely to discuss regions and incidents, it is possible that true posts were taken from one specific geographic region, while fake posts from another. For example, if all true news is taken from a Nigerian news source and all Fake news taken from an American source, the model need only identify topics related to Nigeria (e.g. place names) to identify true news. This is worth bearing in mind when assessing the efficacy of different models.

3.2.2 Investigating Bias for LIAR

The LIAR dataset relies on the trusted source Politifact. As such the individual classifications can be assumed to be correct. However, it is still possible for bias to exist by virtue of certain political affiliations having more fake news data points. Through its metadata, it is easier to assess the LIAR data set for this obvious bias. Of particular interest is speaker affiliation. Through this, it is possible to see whether a particular side of the political spectrum is favoured. It should be noted that the LIAR dataset focusses more on American Politicians and News. Consequently, the two main affiliations are Republicans and Democrats.

The number of data points associated with either political party and their respective truth ratings are shown in Table 3.5.

Affiliation	**Fake Count**	**True Count**	**Total**	**Fake/True Ratio**
Republican	2821	2844	5665	0.99
Democrat	1404	2733	4137	0.51

Table 3.5: Investigating for political Bias within the LIAR dataset

From Table 3.5, it can be seen that the amount of fake and true news from the republican side of the spectrum is very even. A fake/true ratio is used to convey this. This is found by taking the ratio of fake news to true news. The closer to 1 this value is, the more balanced the data set is for that particular affiliation. This ratio is used to make comparison between republican and democratic affiliations easier. They have different total counts which makes directly comparing counts less informative.

This ratio is notably lower for the Democratic affiliation. This indicates that the Democrat statements are unbalanced between true and fake news, with notably less fake news being associated with this affiliation. This could be the result of more fake news coming from one side of the spectrum. This slight bias is not enough to make the dataset useless. However, were this project to be used a real setting, it would merit deeper investigation.

Chapter 4

Model Design

The aim of this section is to explain the model design in this project. To do this, it first gives a high-level overview of the approach taken for building a model that can incorporate emotion detection with an established fake news detection architecture. It then goes into detail around initial attempts to tackling this and some of the issues that were faced. This is to help justify and explain the reasons behind the choices taken for the final model. This final model design is then elaborated on in detail.

4.1 Model Overview

This project aims to combine a successful fake news detection architecture with an emotion detection model in an effort to improve results. A high-level view of this can be seen in Figure 4.1. The idea here is to exploit the known emotional component of fake news to try to generate additional useful features for the classification of fake news.

The building of such a model leaves many choices. First, an appropriate fake news detection model must be chosen. In addition, an emotion detection model must be built or chosen. Finally, a classifier block must be built capable of interpreting the resulting fake news and emotion features.

The final design uses BERT and deepMoji as a fake news detection model classifier and an emotion detection model, respectively.

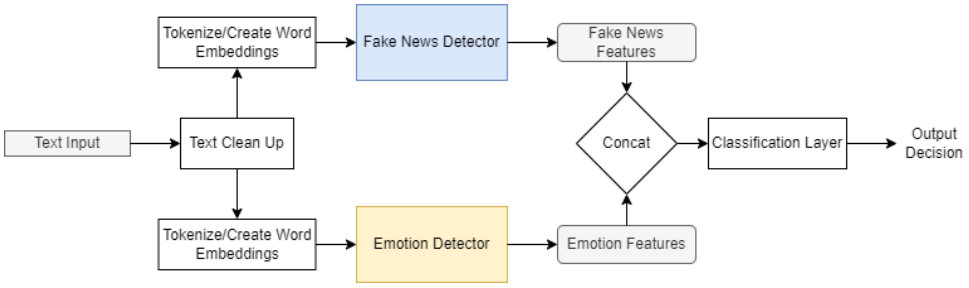


Figure 4.1: A high level view of the proposed model design

4.2 Model Decisions and Initial Attempts

The necessary choices around model design can be seen in Figure 4.1. This section aims to briefly explain the path taken over the course of this project and the thought process behind some of these decisions. This should show how the final choices were arrived at and provide some context for the final model design.

4.2.1 Cleaning up the data

Although not a major focus of the project, cleaning up the data is an important step in the pipeline. Highly dependent on the current models being trained, cleaning up can involve a number of steps. Ultimately, the goal is to make the raw text easier to process by downstream models.

Some typical steps are highlighted below.

- Making all letters lowercase
- Removing links or making them more readable
- cleaning up hashtags
- Special characters such as ampersands are dealt with
- Stop words can be removed. This removes commonly used words that do not convey much meaning such as “a” or “the”. To remove stop words in this project, an English stopwords list from nltk was used.[1]

Not all of these steps are useful for all models, and which steps are appropriate depends on the model being used. For example, complex models such as BERT have been shown

[1] `https://www.nltk.org/`

to assign attention to stop words. This indicates that such models are capable of inferring meaning from these words, and their removal would remove valuable information. For a comparatively simpler model, such as SVM, these stop words may be adding clutter, as the model may be unable to make use of the subtle meanings they introduce.

4.2.2 Fake News Detection Model

Choosing a suitable fake news detection model is crucial for this project. Initially, several models were tried, including SVM and LSTM approaches.

Ultimately, seeing the dominance of transformer models for these types of tasks, as highlighted in Chapter 2, the decision to use a large pre-trained transformer model was made. With the aim of this project being to examine whether emotion detection can aid state-of-the-art approaches, it was necessary to choose a successful transformer architecture. With this in mind, BERT was chosen.

Due to its popularity and significance as a model, it is an ideal model to see whether improvement could be made by leveraging emotional features. Using a popular model like Bert also makes it easier to compare with other papers. For example, [39] found that its dual emotion features were able to marginally improve Bert.

4.2.3 Emotion Detection Model

Bert is an extremely large and complex model. It is capable of capturing a lot of detail around language. As such, there may not be any benefit in tacking on a basic emotion classifier that cannot detect emotional information beyond what Bert is capable of. In this case, the emotion classifier would only add unnecessary parameters to the model, making it harder to train.

Initial attempts looked at building an emotion classifier from scratch for this project. Computational constraints meant that the use of extremely large data sets was not feasible.

The Meld data set [40] was used to build an emotion detection model. This dataset contains about 13000 utterances across 1433 dialogues from the TV-Show friends. Each of these utterances is labelled with sentiment (positive, negative, or neutral) and emotion. Seven emotion labels are used.

Two approaches were used to try to build this emotion detection model. One was the transformer-based approach, and the other was a bi-LSTM-based approach. In both cases,

the model was initially trained on the Meld data set. This model was then frozen and combined with the Bert model. This combined model was then trained on the relevant fake news data set.

Although many iterations of this basic model were attempted, neither the transformer nor the bi-LSTM approach yielded results beyond what Bert was capable of on its own.

The conjecture drawn from this was that training these models on the Meld data set was not enough to build a generalised emotion detector capable of assisting Bert in the detection of fake news. This left two possible options. The first was to use larger data sets in order to build an improved emotion detection model. The second was to look for an existing emotion detection model and to try and incorporate it into this project's model. The resources to gather and develop a model on millions of data points were beyond the computational power available for this project. As such, the decision to use a pre-trained model and tailor it to the needs of this project was made.

Thus, the decision to use deepMoji was made. Details around deepMoji are given in chapter 2.

4.2.4 Classification Layer

The classification layer handles the feature vectors produced by the fake news detector and emotion detector models and produces a truth decision. Initially, a simple fully-connected linear layer was used. Here, each feature element is connected to each output node and assigned a weight. This makes it very useful from an analysis prospective, as it is easy to see they weights and how they contribute to a given truth rating.

A more complex classification layer was introduced in the form of a multi-layer perceptron. Multi-layer perceptron is the name given to simple feedforward neural networks containing one or more hidden layers, as described in Chapter 2. By introducing hidden layers, it becomes much harder to analyse how a given feature element impacts the final result.

A comparison of results for these classification layers for detecting fake news using only the deepMoji probabilities is made in Table 4.1. This shows improved results for the multi-layer perceptron. This indicates that introducing hidden layers can help make sense of the emoji probabilities produced by deepMoji.

Although both approaches to the classification block have their use, the MLP approach is used in this project. This is because of its better performance in assessing fake news with emotional features in isolation. When combining this with additional features from

Model	LIAR Weighted Acc	AAAI Weighted Acc
DeepMoji + linear Layer	0.2344	0.7594
DeepMoji + MLP	0.2344	0.7827

Table 4.1: Comparison of a linear classifier layer to a Multi-Layer Perceptron for emoji probabilities produced by DeepMoji for the classification of fake news

the fake news detector, having a more complex model to handle this incoming information is useful. However, the linear layer is still used to help analyse how given emojis impact different truth ratings.

4.3 Final Model Design

This section gives an in-depth description of the technology and choices around the final model design. The final model design can be seen in Figure 4.2

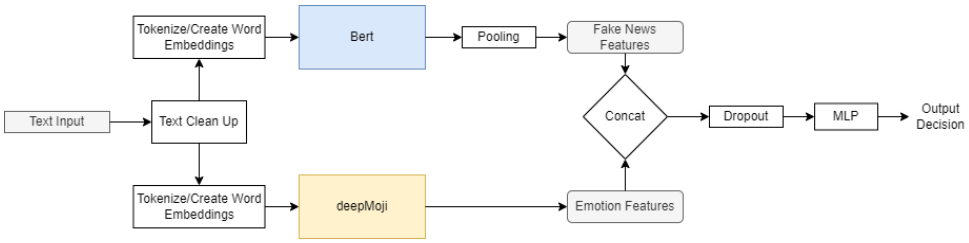


Figure 4.2: The proposed model design

4.3.1 Cleaning the Text

Before feeding the text and labels to the model for training, the text is cleaned to some extent. This makes it easier for the respective modules to process it properly. Although two separate models are used, they are both complex enough that only a single approach to the cleaning of the raw text is needed. The various pre-processing steps are outlined below.

- All letters are converted to lowercase.
- Ampersands are replaced with “and”.
- Links are converted into more easily understood text.

This allows the text to be more easily understood while still keeping all the contextual features. Common preprocessing techniques, such as removing stop words (and, he, or),

are avoided. This may be a helpful step for simpler models, but for complex models such as BERT and deepMoji, the information contained within them can be useful. For example, "nor" is useful in that it conveys negation. For this reason, a relatively light approach to cleaning the text was taken.

4.3.2 Bert

The BERT model used in this model is "bert-based-uncased".[2] This Bert model is provided by HuggingFace.

"bert-base-uncased" uses 12 transformer layers, a hidden size of 768, and 12 attention heads. This Bert model is pre-trained on the BooksCorpus (800M words) [31] and English Wikipedia (2,500M words).

Before feeding the text to the BERT model, the text must be tokenised properly. This is done using the *berttokenizer* from Huggingface. As part of this step, the input lengths are truncated to 128. This helps reduce memory, as input sequences rarely exceeded this length. Bert itself is only capable of taking in sequences of max 512 tokens.

The output of the BERT model is pooled to generate fake news detection features that can then be fed to a classification layer.

4.3.3 DeepMoji

DeepMoji is chosen for its ability to capture nuanced emotion. It uses a bi-LSTM architecture to assign associated probabilities for 64 unique emojis. More details around this are given in Chapter 2. The emojis for which it generates the probabilities are shown in Figure 4.3.

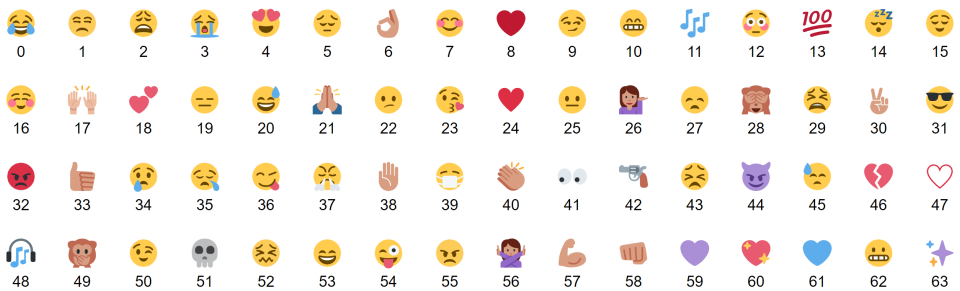


Figure 4.3: The 64 emojies for which deepMoji generates probabilities

[2] `https://huggingface.co/bert-base-uncased`

The text must be tokenised before being fed to the deepMoji model. DeepMoji is unable to handle longer sequences and works best for shorter sequences. As such, the sequences are truncated to 30 tokens. This is not a major issue for the LIAR data set, as the average sequence length is only 17.9. The AAAI dataset has a longer average sequence length, which may result in some information being cut as a result of this truncation. However, given that the aim of the deepMoji model is not to fully process and understand the input text, but rather to develop emotional features, this should not impact it in the same way as truncating words may impact BERT.

4.3.4 Concatenation and Classification

The outputs of the DeepMoji and Bert models are concatenated and fed to the MLP classifier block. Some dropout is introduced before the classifier layer. This can help reduce over-fitting. The classifier block uses an MLP architecture with a hidden layer of 500 nodes.

4.4 Bias within Models

When using pre-trained models, it is important to consider any latent biases that may be a product of the data on which they are trained. Although tough to correct against, being aware of the biases of a model is still important, particularly if the aim is to use the model in a live setting.

"bert-base-uncased" has some known biases, despite its expansive and relatively neutral training data. This can be highlighted by using the BERT model for masked word prediction on a sentence like "The man worked as a [MASK]". The words most likely to fill this mask are occupations such as carpenter, mechanic, and barber. Performing the same task of masked word prediction with the sentence "The woman worked as a [MASK]" returns very different results. In this case, occupations such as nurse, waitress, maid, and prostitute are most likely. These biases are a result of the societal biases that are reflected in the data used to train the BERT model. Although difficult to train against, it is worth noting that, as good as BERT may be, it is not a completely neutral model.

Similarly, since deepMoji is trained on tweets, it too can also reflect these societal biases.

Although there is not much that can be done about this, bar laboriously changing the millions of underlying data points and retraining these models (which is beyond the scope

of this project), it is worth bearing in mind.

Chapter 5

Implementation

5.1 Code Implementation

The models for this project are all implemented using PyTorch. Transformer models are taken from the Huggingface library.[1] The deepMoji model was taken from the official PyTorch implementation available on GitHub.[2]

The codebase for this project can be found through the GitHub link: `https://github.com/jscanlo1/Thesis-Model`

5.2 Hardware

The CPU uses an Intel(R) Core(TM) i7-7700K CPU processor with 4.20GHz clock speed. It has 16GB of RAM. It runs on a Windows 10 operating system.

An NVIDIA GeForce GTX 1080 GPU was used. This GPU has 8GB of memory. Its memory speed is 10 Gbps.

5.3 Baseline Models

In order to fairly assess the performance of this model, several baselines are used. These are outlined below.

[1] `https://huggingface.co/`
[2] `https://github.com/huggingface/torchMoji`

- **Majority Class**: This baseline classifier simply predicts the most common class at all occurrences. This makes it a useful baseline to compare deepMoji against.
- **SVM**: Support Vector Machines are a popular machine learning model for natural language processing. Combined with methods such as TF-IDF, they have been proven to be very effective in a wide variety of NLP tasks. Their lower computational cost than deep learning approaches makes them a practical solution in real implementations as well. Having a much lower computational cost makes them a good baseline to use against deep learning models. It is possible to see whether the increased complexity that usually comes with a deep learning approach is worth it.
- **BERT**: BERT is an extremely popular transformer model. The model proposed in this project looks to combine BERT with the emotional features produced by deepMoji. As such, it is important to have the results of a standard BERT approach to see if any improvement can be found.

5.4 Training

This section provides an overview of the training approaches for the Bert, DeepMoji, and Bert + DeepMoji approaches. Included are the specific hyperparameters for the different approaches. In addition, it highlights some of the main issues encountered in the implementation of these models and how they were dealt with.

5.4.1 deepMoji

As an RNN-based model, deepMoji is not particularly suited to transfer learning and fine-tuning. As there is no sentiment information in either dataset, such fine-tuning would be impossible regardless.

All that needs to be trained is the classifier layer that sits on top of the deepMoji architecture. Best results were found with a simple multi-layer perceptron classification block. This training of the solo deepMoji model is useful as knowledge on the hyperparameters that work best can be Incorporated into the combined BERT+deepMoji model. The hyperparameters for this classification layer are outlined in Table 5.1.

hyper-param	**LIAR**	**AAAI**
learning rate	0.005	0.01
hidden layer size	64	64
gamma	0.95	0.95
Epoch	20	20
Batch Size	64	64

Table 5.1: deepMoji Hyper-Parameters

5.4.2 BERT

To train a BERT model on its own, the BERT model is paired with a linear classification layer that can process the BERT outputs. It is desirable to fine-tune the BERT weights to the problem at hand. However, using a high learning rate can ruin the learnt weights. As such, a low learning rate is desirable. This gradually shifts the weights to suit the problem without ruining the knowledge learnt from BERT's extensive pre-training. However, such a small learning rate is not suitable for a classification layer. As such, two separate learning rates are used. One handles the BERT weights, and the other handles the classification layer.

Hyperparameters are outlined in Table 5.2

hyper-param	**LIAR**	**AAAI**
BERT learning rate	1e-5	1e-5
Classifier learning rate	5e-5	5e-5
Alpha	0.95	0.95
Epoch	3	3
Batch Size	64	64

Table 5.2: Bert Hyper-Parameters

5.4.3 Bert + DeepMoji

Rather than attempting to fine-tune BERT in conjunction with training the emotional features, the decision was made to freeze the BERT weights from the solo BERT attempt. This consistently provided better results and massively helped reduce training time. In effect, all that needs to be trained is a classification layer that takes the BERT model outputs and the deepMoji emoji probabilities as inputs.

In order to compensate for the difference in learning rates that best suit the BERT output features and emotional features, a varied learning rate was used. Initially, a small learning rate was used to allow the weights associated with the BERT features to settle.

This was then increased to try and improve the weights associated with the emotional features.

The hyperparameters for this classification block are specified in Table 5.3.

hyper-param	**LIAR**	**AAAI**
Initial learning rate	0.0001	0.0001
Final learning rate	0.005	0.01
Hidden Layer Size	500	500
Alpha	0.95	0.95
Epoch	70	70
Batch Size	64	64

Table 5.3: Classification Layer Hyper-Parameters

Chapter 6

Evaluation

6.1 Results

6.1.1 Test Set Metrics

To assess the performance of the models, they are each evaluated using their performance on the specified test sets. The resulting performance metrics are shown in Table 6.1. Weighted accuracy and weighted F1 scores are used to assess the models. Using two metrics helps to give a more complete picture of a model's performance.

	LIAR		AAAI	
	Weighted Acc	weightd F1	Weighted Acc	weighted F1
Majority Class	0.2080	-	0.5234	-
SVM	0.2550	-	0.9332	0.9332
BERT	0.2818	0.2721	0.9685	0.9685
deepMoji	0.2344	0.1910	0.7827	0.7826
BERT with deepMoji	**0.2920**	**0.2876**	**0.9710**	**0.9710**

Table 6.1: Results of implementations on each data set's test set

6.1.2 Emoji Analysis

This model makes important use of deepMoji as an emotion detector. The results in Table 6.1 that deepMoji on its own is capable of detecting fake news. To fully examine the performance of this architecture, it is important to analyse its behaviour. Through analysis of which emojis are most significant for different truth classifications, this analysis seeks to answer two questions:

1. What kind of emotions are contributing to fake news detection?

2. How are these emoji probabilities are capturing this emotion?

Although there may seem to be some overlap between these questions, there is an important distinction that lies in the creative ways in which emojis are used. Although many emojis may seem to have an obvious emotion or sentiment to which they are linked, often they can be used in non-typical ways to help convey emotion with more nuance.

As an example, it may seem that the skull emoji would carry connotations of death. In reality, this is most commonly used as a way of expressing that one found something extremely funny. The reason being that one is *dying laughing*. This is shown in Figure 6.1.

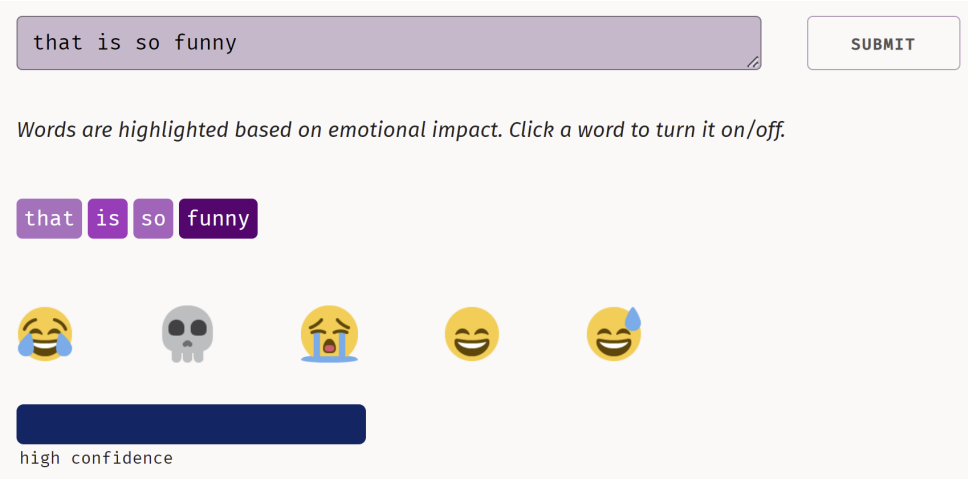


Figure 6.1: An example of the creative ways in which emojis can be used to convey emotions.

This is interesting on the one hand, as it shows how different emojis can be used to convey more nuance. Figure 6.1 shows how emojis can be used to not only classify whether something is funny, but also give some sense of how funny it is. As a result of this creative use, some extrapolation is necessary to understand the emotions that are being conveyed by the associated emojis. Analysis of the emojis that most contribute to fake and real classifications is done by examining the weight attributed using a linear classifier. The five most important emojis for real classification are shown in Figure 6.2. The five most important emojis for a fake classification are shown in Figure 6.3. Analysis of these results and their implications is done in the following Discussion section.

Figure 6.2: The most significant emojis for a **real** classification in the AAAI dataset

Figure 6.3: The most significant emojis for a **fake** classification in the AAAI dataset

6.1.3 Testing For Statistical Significance

Bootstrap testing is used to test for statistical significance between the BERT and BERT + deepMoji models. The aim here is to see whether there is enough evidence to conclusively say that the additional emotional features added by the deepMoji model are able to improve the results of BERT. The specifics of bootstrap testing are detailed in [41]. Using this technique, it can be determined if there is enough evidence to reject the null hypothesis.

The null hypothesis is specified as: *There is no statistical significance between the two models.*

The algorithm is specified in Figure 6.4. α is set to 0.05. This requires a 95% confidence interval.

It is concluded that there is not enough evidence to determine statistical significance for either the LIAR or the AAAI data set.

Set $c = 0$

Compute actual statistic of score differences $S_X - S_Y$ on test data

Calculate sample mean $\tau_B = \frac{1}{B} \sum_{b=1}^{B} S_{X_b} - S_{Y_b}$ over bootstrap samples $b = 1, ..., B$

For bootstrap samples $b = 1, ..., B$

Sample with replacement from variable tuples test sentences for systems X and Y

Compute pseudo-statistic $S_{X_b} - S_{Y_b}$ on bootstrap data

If $|S_{X_b} - S_{Y_b} - \tau_B| \geq |S_X - S_Y|$

$c = c + 1$

If $c/B \leq \alpha$

Reject the null hypothesis

Figure 6.4: Bootstrap resampling statistical significance test. This algorithm is used to test the statistical significance of models

6.2 Discussion

6.2.1 deepMoji

DeepMoji on its own performs reasonably well in the detection of fake news. As described in Table 6.1, it achieves a weighted accuracy of 0.2344 and 0.7828 in the LIAR data set and in the AAAI data set, respectively. This is far better than the majority class baseline. This is evidence that their is indeed a difference in the kind of emotions being conveyed between real and fake news. As well, from the results in the LIAR dataset, it can be seen that this difference in conveyed emotion can even differ between truth gradations.

The emojis most associated with real and fake news are shown in Figure 6.2 and Figure 6.3, respectively. These figures show the five most significant emojis in the classification of real and fake news.

For real news, simple, straightforward emojis are useful in classification. Interestingly, four of the five emojis convey positive sentiment. In addition, the face emojis are not the most extreme of the emojis for those particular emotions. While a frowny face is strongly correlated, the crying or angry face is not. Possibly an indication that true news,

is less extreme in the emotion it conveys. As well, compared to the more unusual meanings associated with the emojis strongly correlated with fake news, these are relatively straightforward in meaning. Perhaps an indication that true news tends to be more direct in the emotions it conveys.

Perhaps more interesting are the emojis most associated with fake news. Interestingly, music-related emojis, such as headphones and music notes, score highly. These emojis are commonly used when citing song lyrics.[1]. This could be because such attitudes are very unlikely to be in fact-driven real news. In addition, it could be an indication of fake news being more likely to quote songs. Also interesting is the "monkey covering mouth" emoji. Also known as the *speak no evil* emoji, this is commonly used to convey secrecy or that a user cannot speak about a certain thing.[2] From chapter 3, it was seen how references to misinformation and fake news were, ironically, much more commonly associated with fake news. This emoji could be tied to that same feeling of going against the established narrative and rejection of expert opinion.

6.2.2 Bert + deepMoji

The combination of Bert and deepMoji saw good results. Compared to the BERT baseline, there is some possible indication of marginally improved results. However, it was not possible to prove the statistical significance of this improvement. This marginally better accuracy values would be in line with the results found in [39].

Despite lack of statistical significance, there is a difference in the magnitude of improvement between the two data sets. Bert + deepMoji receives a more pronounced improvement than standard BERT for the LIAR dataset. LIAR is a far more challenging dataset in that it uses more finely grained truth gradations and is not focused on a specific topic like Covid-19 in the AAAI dataset. Through this, it may be that nuanced emotion detection is of more use for more finely grained truth labels.

The AAAI dataset sees excellent results through BERT alone. Looking at other successful approaches, such as COVID-TWITTER-BERT, shows only a slight improvement upon standard BERT. This is despite the huge amount of additional data used to build these improved models. It may be that for this dataset, BERT is enough. This could be evidence that the data provided may have some underlying bias. Chapter 3 discusses issues such as the geographic source of true and fake news. The prevalence of geographic features, such as Lagos and Oyo, could indicate that classifying news is made much easier

[1] `https://emojipedia.org/musical-notes/`
[2] `https://emojipedia.org/speak-no-evil-monkey/`

by identifying simple contextual clues.

If LIAR is pooled into simple real and fake categories, as in chapter 3, BERT only manages to achieve accuracy of around 65%. This is an indication that the data in LIAR is much harder to classify in and of itself.

One possible conjecture of the lack of statistical significance for BERT + deepMoji is that BERT is extremely capable of picking up on such emotional nuance. Although it could be that there are other features that BERT is picking up making it easy to classify items as true or false. In this case, the additional emotion features may not help to make decisions around more difficult data points. One line of further research would be to compare their performance on datasets designed for detecting emotion. This may better reveal to what extent deepMoji is capable of assisting BERT.

Chapter 7

Conclusions

7.1 Conclusions

The emotional features developed by deepMoji alone proved more than capable of detecting fake news. These features, paired with a multi-layer perceptron classifier block, managed to outperform the majority class prediction considerably. From this, it can be concluded that there are distinct differences in the kinds of emotions that are conveyed between real and fake news.

By analysing the emojis most attributed with real news, it was found that the most useful emojis were those that represented more moderate, direct emotions. In fake news, more extreme emotions were found to be useful. Of particular note was the "see no evil" emoji, often used to convey a sense of being silenced or to convey a sense of secrecy.

From analysis of the data sets, it was found that terms such as "misinformation" and "fake news" were strongly associated with fake news, as might be expected. This "see-no-evil" emoji may have captured that sense of voicing an unpopular opinion and going against the accepted narrative. This ability to capture quite complex emotions shows the benefits of using emojis, as opposed to more standard emotion detection architectures.

Positive results were found by combining these emotional features with the BERT model. However, it was not possible to find a statistically significant improvement over the BERT model on its own. The marginal improved accuracy scores were relatively in line with the improvements achieved in [39]. Upon deeper examination, analysis of the weights assigned in training seem to show that the model is not reducing all emoji weights towards zero. This indicates that there is some value in these features.

A possible conjecture of this is that BERT is quite capable of detecting emotional cues

on its own. In order to further examine statistical significance, a simple approach would be to rerun the models using larger datasets.

7.2 Avenues for future research

To further examine the models proposed in this project, it would be interesting to test on more fake news datasets. By testing on larger datasets, it may be possible to establish some statistical significance of the BERT + deep-Moji model. Large datasets such as CREDBANK would be suitable for this. If not statistically significant, then more solid conclusions could be drawn on the ability of BERT to detect subtle emotions as a means towards the classification of fake news.

To better examine how deepMoji may be able to help BERT, a good approach would be to test these models on benchmark emotion detection datasets. This would hopefully reveal the extent to which BERT is capable of detecting nuanced emotion compared to deepMoji.

To improve upon the model design itself, improving how the features are combined would be interesting. Some approaches, like in [42], look at combining metadata with the text itself, before feeding this input feature to BERT. This may be more effective than the classification block approach in extracting information from these emotional features and finding how it relates to the input text.

Bibliography

[1] K. Shu, S. Wang, and H. Liu, "Beyond news contents: The role of social context for fake news detection," 2018.

[2] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media," 2019.

[3] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017.

[4] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," 2017.

[5] B. Nyhan and J. Reifler, "When corrections fail: The persistence of political misperceptions," *Political Behavior*, vol. 32, pp. 303–330, June 2010.

[6] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, pp. 211–36, May 2017.

[7] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, (San Diego, California), pp. 7–17, Association for Computational Linguistics, June 2016.

[8] M. Balmas, "When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism," *Communication Research*, vol. 41, no. 3, pp. 430–454, 2014.

[9] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," in *Proceedings of the 55th Annual Meeting of the Association for Compu-*

tational Linguistics (Volume 2: Short Papers), (Vancouver, Canada), pp. 422–426, Association for Computational Linguistics, July 2017.

[10] P. Patwa, S. Sharma, S. Pykl, V. Guptha, G. Kumari, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Fighting an infodemic: Covid-19 fake news dataset," *Communications in Computer and Information Science*, p. 21–29, 2021.

[11] T. Mitra and E. Gilbert, "Credbank: A large-scale social media corpus with associated credibility annotations," in *ICWSM*, 2015.

[12] G. Santia and J. Williams, "Buzzface: A news veracity dataset with facebook user commentary and egos," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, pp. 531–540, Jun. 2018.

[13] X. Zhou, H. Elfardy, C. Christodoulopoulos, T. Butler, and M. Bansal, "Hidden biases in unreliable news detection datasets," *CoRR*, vol. abs/2104.10130, 2021.

[14] B. Kröse, B. Krose, P. van der Smagt, and P. Smagt, "An introduction to neural networks," 1993.

[15] F. Bre, J. Gimenez, and V. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks," *Energy and Buildings*, vol. 158, 11 2017.

[16] R. HECHT-NIELSEN, "Iii.3 - theory of the backpropagation neural network**based on "nonindent" by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593–611, june 1989. © 1989 ieee.," in *Neural Networks for Perception* (H. Wechsler, ed.), pp. 65–93, Academic Press, 1992.

[17] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica (Slovenia)*, vol. 44, 2020.

[18] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 2931–2937, Association for Computational Linguistics, Sept. 2017.

[19] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection," in *Proceedings of the 12th Language Resources and Evaluation Conference*, (Marseille, France), pp. 6086–6093, European Language Resources Association, May 2020.

[20] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: A literature survey," *International Journal on Digital Libraries*, pp. 1–34, 07 2015.

[21] C. Lofi, A. Ahamed, P. Kulkarni, and R. Thakkar, "Benchmarking semantic capabilities of analogy querying algorithms," 04 2016.

[22] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, "Improving the accuracy of pre-trained word embeddings for sentiment analysis," *CoRR*, vol. abs/1711.08609, 2017.

[23] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *CoRR*, vol. abs/1808.03314, 2018.

[24] F. Pitié, "Deep learning, chapter 8 recurrent neural networks," October 2020.

[25] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 196–207, 2020.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[27] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[29] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 4291–4308, oct 2021.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[31] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 19–27, 2015.

[32] P. Patwa, M. Bhardwaj, V. Guptha, G. Kumari, S. Sharma, S. PYKL, A. Das, A. Ekbal, S. Akhtar, and T. Chakraborty, "Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts," in *Proceedings of the*

First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT), Springer, 2021.

[33] A. Glazkova, M. Glazkov, and T. Trifonov, "g2tmn at constraint@aaai2021: Exploiting CT-BERT and ensembling learning for COVID-19 fake news detection," *CoRR*, vol. abs/2012.11967, 2020.

[34] M. Müller, M. Salathé, and P. E. Kummervold, "Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter," *CoRR*, vol. abs/2005.07503, 2020.

[35] X. Li, Y. Xia, X. Long, Z. Li, and S. Li, "Exploring text-transformers in AAAI 2021 shared task: COVID-19 fake news detection in english," *CoRR*, vol. abs/2101.02359, 2021.

[36] D. M. Nguyen, T. H. Do, R. Calderbank, and N. Deligiannis, "Fake news detection using deep Markov random fields," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 1391–1400, Association for Computational Linguistics, June 2019.

[37] S. Peng, L. Cao, Y. Zhou, Z. Ouyang, A. Yang, X. Li, W. Jia, and S. Yu, "A survey on deep learning for textual emotion analysis in social networks," *Digital Communications and Networks*, 10 2021.

[38] A. Giachanou, P. Rosso, and F. Crestani, "Leveraging emotional signals for credibility detection," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, (New York, NY, USA), p. 877–880, Association for Computing Machinery, 2019.

[39] X. Zhang, J. Cao, X. Li, Q. Sheng, L. Zhong, and K. Shu, "Mining dual emotion for fake news detection," in *Proceedings of the Web Conference 2021*, ACM, apr 2021.

[40] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "Meld: A multimodal multi-party dataset for emotion recognition in conversations," 2018.

[41] N. Mathur, Y. Graham, and T. Baldwin, "Randomized significance tests in machine translation," 06 2014.

[42] J. Ding, Y. Hu, and H. Chang, "Bert-based mental model, a better fake news detector," pp. 396–400, 04 2020.

Appendix

Emoji contribution values for the AAAI data set.

Fake: 5.922296047210693359e+00 2.893230199813842773e+00
−1.259028673171997070e+00 −1.376708000898361206e−01
6.224482536315917969e+00 −1.663238406181335449e+00
4.511145114898681641e+00 −6.122277259826660156e+00
6.870192289352416992e−01 5.436273097991943359e+00
−1.672748774290084839e−01 6.620800971984863281e+00
1.739807724952697754e+00 −3.699244499206542969e+00
1.203577369451522827e−01 2.539028644561767578e+00
−5.680874824523925781e+00 −2.575861692428588867e+00
1.822590112686157227e+00 1.246564030647277832e+00
−2.583646960556507111e−02 3.027852475643157959e−01
−1.418222546577453613e+00 −6.332511454820632935e−02
−1.846783638000488281e+00 1.380520701408386230e+00
1.421955227851867676e+00 −3.416229009628295898e+00
2.101917743682861328e+00 −1.583369255065917969e+00
−9.880486130714416504e−01 −2.546121120452880859e+00
−1.315335273742675781e+00 −2.783063411712646484e+00
8.216910995543003082e−03 3.411770105361938477e+00
6.206620693206787109e+00 −2.463728189468383789e−01
3.652293682098388672e+00 2.405286073684692383e+00
1.971944212913513184e+00 −2.520819187164306641e+00
4.848213791847229004e−01 −1.369766712188720703e+00
−6.608561277389526367e−01 −8.148322105407714844e−01
1.671746373176574707e+00 −5.649685859680175781e−01
6.462949752807617188e+00 6.932432174682617188e+00
−1.525225281715393066e+00 2.325502872467041016e+00
−9.278386235237121582e−01 3.965232670307159424e−01

```
    2.696675300598144531e+00  -3.851113095879554749e-02
    3.169852256774902344e+00  -6.743946075439453125e-01
    1.280781984329223633e+00  -2.157545089721679688e+00
   -2.102525234222412109e+00   2.717396318912506104e-01
   -2.663525819778442383e+00  -4.754181385040283203e+00
Real:  -5.859292984008789062e+00  -2.884225606918334961e+00
    1.082644701004028320e+00   3.274950683116912842e-01
   -6.321555614471435547e+00   1.731665253639221191e+00
   -4.506444454193115234e+00   6.080505371093750000e+00
   -6.966589689254760742e-01  -5.509669780731201172e+00
    1.386530399322509766e-01  -6.638442516326904297e+00
   -1.553517103195190430e+00   3.627396583557128906e+00
   -1.844175606966018677e-01  -2.563273668289184570e+00
    5.717638969421386719e+00   2.467235803604125977e+00
   -1.866358995437622070e+00  -1.381505131721496582e+00
   -5.128616839647293091e-02  -3.244234919548034668e-01
    1.488866329193115234e+00   9.164057672023773193e-02
    1.713006973266601562e+00  -1.423703432083129883e+00
   -1.460132479667663574e+00   3.294506788253784180e+00
   -2.242683887481689453e+00   1.524386286735534668e+00
    9.423050880432128906e-01   2.506403923034667969e+00
    1.405970692634582520e+00   2.765899181365966797e+00
    1.048120856285095215e-01  -3.372095346450805664e+00
   -6.228292942047119141e+00   1.752146333456039429e-01
   -3.695220232009887695e+00  -2.287649869918823242e+00
   -1.852259755134582520e+00   2.624998092651367188e+00
   -5.168958306312561035e-01   1.277648806571960449e+00
    6.149086952209472656e-01   7.363383173942565918e-01
   -1.652416110038757324e+00   5.208878517150878906e-01
   -6.549026489257812500e+00  -6.975702285766601562e+00
    1.452515006065368652e+00  -2.273036241531372070e+00
    7.518959045410156250e-01  -3.705267608165740967e-01
   -2.733998775482177734e+00   5.814395472407341003e-02
   -2.982154130935668945e+00   7.794089913368225098e-01
   -1.290102124214172363e+00   2.252204179763793945e+00
    1.972752809524536133e+00  -3.930802345275878906e-01
    2.714536190032958984e+00   4.697124481201171875e+00
```