Fake News Detection and Production using Transformer-based NLP Models

Authors: Branislav Sándor - Student Number: 117728
Frode Paaske - Student Number: 102164
Martin Pajtás - Student Number: 117468

Program: MSc Business Administration and Information Systems – Data Science
Course: Master’s Thesis
Date: May 15, 2020
Pages: 81
Characters: 184,169
Supervisor: Daniel Hardt
Contract #: 17246
Acknowledgements

The authors would like to express their sincerest gratitude towards the following essential supporters of life throughout their academic journey:

- Stack Overflow
- GitHub
- Google
- YouTube
- Forno a Legna
- Tatra tea
- Nexus
- Eclipse Sandwich
- The Cabin Trip
- BitLab

“Don’t believe everything you hear: real eyes, realize, real lies”
Tupac Shakur

“A lie told often enough becomes the truth”
Vladimir Lenin
Glossary

The following is an overview of abbreviations that will be used in this paper.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>Bidirectional Encoder Representations from Transformers</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>Bidirectional Long Short-Term Memory</td>
</tr>
<tr>
<td>BoW</td>
<td>Bag of Words</td>
</tr>
<tr>
<td>CBOW</td>
<td>Continuous Bag of Words</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>CV</td>
<td>Count Vectorizer</td>
</tr>
<tr>
<td>EANN</td>
<td>Event Adversarial Neural Network</td>
</tr>
<tr>
<td>ELMo</td>
<td>Embeddings from Language Models</td>
</tr>
<tr>
<td>FFNN</td>
<td>Feed-Forward Neural Network</td>
</tr>
<tr>
<td>GPT-2</td>
<td>Generative Pretrained Transformer 2</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>KNN</td>
<td>K-Nearest Neighbors</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short-Term Memory</td>
</tr>
<tr>
<td>MLM</td>
<td>Masked Language Modeling</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NSP</td>
<td>Next Sentence Prediction</td>
</tr>
<tr>
<td>OOV</td>
<td>Out of Vocabulary Token</td>
</tr>
<tr>
<td>PCFG</td>
<td>Probabilistic Context-Free Grammars</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>RST</td>
<td>Rhetorical Structure Theory</td>
</tr>
<tr>
<td>SEO</td>
<td>Search Engine Optimization</td>
</tr>
<tr>
<td>SGD</td>
<td>Stochastic Gradient Descent</td>
</tr>
<tr>
<td>SVM</td>
<td>Support-Vector Machine</td>
</tr>
<tr>
<td>TF</td>
<td>Term Frequency</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>Term Frequency - Inverse Document Frequency</td>
</tr>
<tr>
<td>UNK</td>
<td>Unknown Token</td>
</tr>
<tr>
<td>VRAM</td>
<td>Video RAM</td>
</tr>
</tbody>
</table>
Abstract

This paper studies fake news detection using the biggest publicly available dataset of naturally occurring expert fact-checked claims (Augenstein et al., 2019). Based on existing theory that defines the task of fake news detection as a binary classification problem, this paper conducted an extensive process of reducing the label space of the dataset. Traditional machine learning models and three different BERT-based models were applied to the binary classification task on the data to investigate the performance of fake news detection. The RoBERTa model performed the best with an accuracy score of 0.7094. This implies that the model is capable of capturing syntactic features from a claim without the use of external features. In addition, this paper investigated the feasibility and effects of expanding the existing training data with artificially produced claims using the GPT-2 language model. The results showed that the addition of artificially produced training data, whether fact-checked or not, generally led to worse performance of the BERT-based models while increasing the accuracy scores of the traditional machine learning models. The paper finds that the Naïve Bayes model achieved the highest overall score on both the fact-checked and non-fact-checked artificially produced claims in addition to the human-produced training data, with accuracy scores of 0.7058 and 0.7047, respectively. These effects were hypothesized to be caused by differences in the underlying architecture of the different models, particularly the self-attention element of the Transformer architecture might have suffered from the stylistic and grammar inconsistencies in the artificially produced text. The results of this paper suggest that the field of automatic fake news detection requires further research. Specifically, future work should address the lack of sufficient data quality, size, and diversity, the increasing demand for computational resources, and inadequate inference speed severely limiting the application of BERT-based models in real-life scenarios.

Keywords: Fake News, Transformers, Natural Language Processing, BERT, GPT-2, Fake News Detection,
Table of Contents

Acknowledgements ............................................................................................................. 2
Glossary ................................................................................................................................. 3
Abstract ................................................................................................................................ 4
Introduction ............................................................................................................................. 7
  Natural Language Processing ................................................................................................. 8
  Motivation .............................................................................................................................. 9
    Economics of Fake News ...................................................................................................... 9
Research Aim .......................................................................................................................... 13
Research Question .................................................................................................................. 13
Theoretical Framework .......................................................................................................... 15
  News ..................................................................................................................................... 15
    Fake News .......................................................................................................................... 17
  Fake News Detection ............................................................................................................ 20
    Digital Media and Fake News ............................................................................................ 21
    Feature Extraction ............................................................................................................. 23
    Model Creation .................................................................................................................. 25
    Datasets ............................................................................................................................. 27
Related Work .......................................................................................................................... 29
  Text Features From Claims ................................................................................................. 29
  Fake News Detection Using Transformer Models .............................................................. 31
  Artificial Fake News .......................................................................................................... 32
  Limitations of Existing Research ...................................................................................... 33
Methodology ........................................................................................................................... 35
  Research Philosophy .......................................................................................................... 35
    Ontology ............................................................................................................................ 35
    Epistemology ..................................................................................................................... 35
    Methodological Choice ...................................................................................................... 36
Data ....................................................................................................................................... 38
  Primary Data ....................................................................................................................... 38
  Data Preparation and Pre-processing ................................................................................... 39
  Label Reduction ................................................................................................................... 40
Text Pre-processing ................................................................................................................ 43
  Text Feature Extraction Techniques .................................................................................... 45
    N-Grams ............................................................................................................................. 46
    Bag-of-Words ...................................................................................................................... 46
    Term Frequency – Inverse Document Frequency .............................................................. 47
Evaluation Metric .................................................................................................................... 67
  Accuracy Score ..................................................................................................................... 67
Model evaluation ................................................................................................................... 67
Model Optimization ................................................................................................................ 68
Creation of Artificial Text Using GPT-2 .................................................................................. 68
  Machine Produced Text ........................................................................................................ 68
  Fact-Checking ....................................................................................................................... 70
  Fact-Checking Artificial Claims ........................................................................................... 71

Results .................................................................................................................................. 72
  Results on Natural Claims Dataset ...................................................................................... 74
  Results on Natural Claims Dataset and Non-Fact-Checked Artificial Claims ..................... 74
  Results on Natural Claims Dataset and Fact-Checked Artificial Claims .............................. 75

Discussion .............................................................................................................................. 77
  Limitations ........................................................................................................................... 82
  Future Work ......................................................................................................................... 83
  Reflections ............................................................................................................................ 85

Conclusion .............................................................................................................................. 89

Bibliography ........................................................................................................................... 90

Appendix ................................................................................................................................ 102
  Appendix 1 – Label reduction of MultiFC Dataset ............................................................... 102
  Appendix 2 – Optimal GridSearch Hyperparameters for Baseline Models ......................... 102
  Appendix 3– Manually Verified and Labeled Outputs From GPT-2 ...................................... 102
Introduction

Due to technological advancements in media and communication, people nowadays have access to large amounts of information provided by an enormous number of sources. While this allows consumers to access relevant information in a fast and cost-efficient manner, it has created an environment where fake news proliferates. The potential negative effects of fake news have come to the attention of the globalized world due to recent socio-political events such as the 2016 US Presidential election. Therefore, fake news detection has gained increasing popularity and relevance among researchers and the public. With the emergence of revolutionary model architectures in natural language processing, this sub-field of artificial intelligence has been employed to find solutions to the problem of effectively and efficiently detecting fake news.

This paper will study the application of BERT-based models built on the Transformer architecture on the fake news detection task using the biggest publicly available dataset of naturally occurring, fact-checked claims from numerous fact-checking websites. Moreover, this paper will investigate the impact of utilizing artificially produced text as additional training data on the performance of traditional machine learning models and modern deep learning models. Additionally, this paper provides a discussion of the challenges and opportunities associated with implementing the selected models and methods for the task of fake news detection.
Natural Language Processing

Humans are the most intelligent species on earth. Our ability to talk and understand each other by utilizing natural language allows us to effectively communicate and share information. Natural language develops gradually over long periods of time and is used by humans and can take form as speech or text. Mandarin, Spanish, English and Hindi-Urdu are currently considered the most widespread natural languages (Hammond, 2019). Unlike the mentioned languages, human-constructed languages, also referred to as artificial (e.g. computer languages, Esperanto) are not regarded as natural languages (Lyons, 1991). Natural language consists of sets of rules that limit and shape a specific grammar for a given language (Nordquist, 2020). These rules ensure smooth interpretation of a certain opinion, event or emotion. Despite natural language being governed by rules, speech and text can hardly be referred to as structured data (Lopez Yse, 2019). Unstructured data are difficult to store, manipulate and derive meaning from, thus, machines are not capable of processing and understanding natural language in its raw, unprocessed form. However, a multidisciplinary sub-field of artificial intelligence and linguistics, called Natural Language Processing (NLP) emerged to address this issue.

The main purpose of NLP is to enable computers to understand natural language. Due to traits such as high complexity, long distance dependencies and ambiguous words, NLP performed by a machine constitutes a complex task. Multiple methods and techniques are required to successfully process natural language in verbal or written form. However, the benefits that NLP offers make this field appealing to entities from a broad variety spheres such as Google, Apple or the US department of defense (Eggers, Malik, & Gracie, 2019).

Technical advancement has propelled computers to a level at which they can perform some tasks faster or even better than humans, this is also the case for tasks related to natural language. Modern computers are capable of processing large amounts of data in a shorter time than any human. Therefore, by successfully understanding natural language, machines can speed up, improve or automate tasks such as text classification, sentiment analysis, question answering, information extraction, text generation, etc. The ability to automate these tasks enables many use cases for NLP for both beneficent and malicious actors.
Early attempts at NLP related tasks date back to the development of the Turing test in 1950 (Canuma, 2019). The Turing test is a test of machines’ ability to exhibit human-like intelligent behavior (Canuma, 2019). Early NLP systems and solutions were based on large and complex sets of hand-written-rules that enabled machines to understand natural language (Canuma, 2019). Examples of NLP systems based on these rules are machine translators or simple chatbots (Arya, 2019; Hutchins, 2005). Despite the often-high complexity of these rules, this approach was not ideal since clearly defined rules are incapable of capturing all intricacies of natural language, as natural language develops continuously. The emergence of machine learning in mid 1980s coupled with improvements to computer hardware influenced the field of NLP which resulted in a slow refraining from the utilization of hand-written rules in NLP (Canuma, 2019). The adoption of machine learning methods changed NLP by enabling it to use statistical reasoning to detect patterns in languages from analysis of large text corpora (Canuma, 2019).

Motivation

Despite the potential dangerous consequences of intentional misinformation being raised by the World Economic Forum as early as 2013, it was not until 2016 that the phenomenon of fake news started receiving significant attention (Charlton, 2019; World Economic Forum, 2013). 2016 was a landmark year for surprising socio-political changes, specifically the outcome of the US presidential election and the Brexit referendum in the UK, both of which became controversial due to involvement of Cambridge Analytica, a data analytics company that came under the spotlight for its unethical processing of personal data (Scott, 2018). To emphasize the sociopolitical as well as economic significance of fake news, Oxford Dictionary’s 2016 Word of the Year was ‘post-truth’, which was defined as: “relating to or denoting circumstances in which objective facts are less influential in shaping public opinion than appeals to emotion and personal belief” (Oxford University Press, 2016). This served as a proof of the transition to the next stage of the information age that the globalized world had undertaken (Charlton, 2019).

Economics of Fake News

To understand why and how fake news has become serious issue, it is important to first understand the digital transformation that news publishing and distribution have undergone. Historically, the business of news publishing leveraged vertical integration to control the supply chain in terms of both
the production and the distribution of news, mainly through printing and selling newspapers, thus adopting a linear business model (Martens, Aguiar, Gomez-Herrera, & Mueller-Langer, 2018).

However, with advancements in digital technologies and the spreading presence of the internet, the cost of news distribution was reduced to near-zero, as the internet became a cheap alternative to printing and delivering physical newspapers (Martens et al., 2018). Furthermore, new technologies enabled online forms of news production, further lowering costs. This also lowered the cost of entry and attracted new entrants, such as online-only newspapers, bloggers, social media influencers, etc., which increased competition (Martens et al., 2018). Traditionally, news production was conducted on 24-hour cycle, or one version per day (Martens et al., 2018). However, moving the distribution online meant the possibility to continuously produce and update news throughout the day (Martens et al., 2018). Shorter production cycles, partly amplified by increasing competition, have had significant impact, as it substantially reduced the time allocated for fact-checking and quality control (Martens et al., 2018). Another consequence of strong competition is duplication or increased numbers of substitutes in terms of reported news by multiple news producing outlets, which may result in consumers being reluctant to pay for news that can be accessed elsewhere (Martens et al., 2018). This issue might be addressed by differentiation, although most news sites opted for ‘freemium’ models, offering free access to a number of articles and hiding the rest behind a paywall (Martens et al., 2018). Additionally, compared to selling a physical newspaper as a bundle of articles, news sites now offer the readers an option to choose just the articles they find interesting, increasing the relevance of the news for the consumers (Martens et al., 2018).

How readers access news articles changed with the advent of search engines partly as a solution to help consumers navigate increasing amounts of information available on the internet (Martens et al., 2018). Search engines, being a multi-sided platform matching news consumers with news producers, became the mediators of news consumption which meant that the function of news curation historically performed by news editors now transferred to algorithm-driven search engine platforms (Martens et al., 2018). Subsequently, the ranking of news articles, previously decided by the editors of a newspaper (e.g. which news article will be on the front page), has become delegated to a ranking algorithm as a part of a search engine (Martens et al., 2018). However, search rankings can be influenced by news producers. This practice is commonly referred to as Search Engine Optimization (SEO) (Martens et al., 2018). This enables malevolent producers of fake news to promote their
content by manipulating search engine results (Martens et al., 2018). Notably, advertising revenue plays a role in search rankings, as both search engines and news publishers, whose webpages contain ads, stand to profit (Martens et al., 2018). Nonetheless, while news publishers need to protect their branding and their market positioning, search engines can adapt a more advertising-related profit approach, given that they provide relevant news to consumers (Martens et al., 2018).

The emergence of social media represented the next step in the transformation of news distribution (Martens et al., 2018). Having attracted a vast number of users to their platforms, as well as successfully managing to keep them engaged there, news publishers followed suit and created a presence on these platforms where their readers spend increasing amounts of time (Martens et al., 2018). Shearer and Gottfried (2018) found that 68% of US adults consume news on social media at least occasionally. Social media apps like Facebook and Instagram have become important intermediaries for news consumption as they offer more convenience for users on mobile devices (Martens et al., 2018). Specifically, social media differ from search engines in enabling interaction among networked users on the platform, which not only affects what news an individual receives, but also enables propagation of news throughout the network via posting and sharing (Martens et al., 2018). As with search engines, social media platforms utilize an algorithm-driven approach to selecting news content for users with the goal of generating and maximizing interactions. This in turn drives more traffic that generates more advertising revenue for the platforms (Martens et al., 2018). Additionally, news publishers risk losing even more control over the news content on social media platforms compared to on search engines, as they not only give up control of the distribution but are also faced with the risks of misinterpretation that arise from the ability of users to share and comment on posts (Martens et al., 2018).

As aforementioned, the business of news publishing and distribution used to be based on a vertically integrated linear business model, but following advancements in digital technologies, it has been transformed into an algorithm-driven multi-sided platform business model (Martens et al., 2018). Research from Reuters Digital Report found that two-thirds of web users access news via search engines, social media, or other intermediaries as opposed to directly accessing it on a newspaper’s website (Newman, Fletcher, Kalogeropoulos, Levy, & Nielsen, 2018). These changes in news production and distribution as well as news consumption have several consequences for fake news production and dissemination (Newman et al., 2018).
Firstly, online news distribution decreases the costs of market access for new entrants, which affects the market for news consumption in two significant ways (Martens et al., 2018). On one hand, more entrants leads to participation of news publishers with unknown credibility and veracity of news (Martens et al., 2018). This can result in overall reduction in consumers’ trust towards all news publishers, similar to what Akerlof (1970) described in his seminal paper as market for “lemons”. A recent global study from Ipsos found that fake news prevalence was one of the two main factors that caused trust in traditional media to decline over the past five years (Grimm, Boyon, & Newall, 2019).

On the other hand, lowered costs of market access make it easier and cheaper to produce and disseminate false news (Martens et al., 2018). Kshetri and Voas (2017) found empirical evidence of fake news creators taking advantage of this. Specially, the North Macedonian town of Veles from which more than 140 US politics websites were launched during the one-year period before US presidential elections in 2016 with one teenager earning as much as $16,000 from two pro-Trump websites, exemplifying a viable business opportunity (Kshetri & Voas, 2017). Notably, creators of fake news face little to no risk of legal persecution (Kshetri & Voas, 2017).

Secondly, as the role of the news curator has been taken over by algorithm-driven platforms, their objective of maximizing advertising revenue and online traffic enables trending news, even fake, to rank high and be distributed to large audiences, and in the case of social media platforms to propagate deep and wide into their networks (Martens et al., 2018).

Lastly, it is important to emphasize that algorithm-driven platforms should not bear all the blame, as cognitive biases, such as novelty or confirmation bias, play an essential role in the spread and impact of fake news (Martens et al., 2018). Nevertheless, algorithm-driven news markets have made exploiting these biases easier (Martens et al., 2018).

There have been various approaches to countering Fake News, including establishment of fact-checking mechanisms, both government-organized and crowdsourced, self-regulation by technological companies, reducing financial incentives in advertisement, boosting media literacy and critical thinking skills, and government legislation (Vasu, Ang, Jayakumar, Faizal, & Ahuja, 2018). However, particularly governmental legislation has been controversial as regulating news challenges one of the fundamental values of democracy, the freedom of speech (Tambini, 2017).
Financial incentives to produce and spread fake news coupled with a low risk of punishment due to insufficient or ineffective legal policies, human cognitive bias, and advertising financial incentives for algorithm-driven news curators to propagate popular news have co-created an environment where fake news thrives. However, since the task of curating news has shifted from human editors to algorithms, it is in self-regulation of platforms where fake news can be detected and prevented from further exploiting the favorable economic incentives that have been created due to the digital transformation of news.

Therefore, this paper builds on previous work in automatic fake news detection utilizing machine learning and deep learning models and aims at expanding the current knowledge within the field.

Research Aim
Fake news has enjoyed popularity among a wide variety of academic fields as the topic of interest, in part due to its multidisciplinary nature that makes the research from societal, economical, legal, or technical perspective valuable. This paper aims at developing knowledge in a sub-field of general fake news research, one that studies approaches to fake news detection. This paper will study the performance of traditional machine learning models as well as three different BERT-based models on fake news detection, framed as a binary text classification task. Furthermore, this paper aims at contributing to the existing research field that studies the performance of the aforementioned models solely on the text of claims, which are labelled as true or fake, disregarding external features. In addition, the creation of artificial fake news using the GPT-2 model and its subsequent use as additional training data for the models and the effects on the performance of models will be investigated.

Research Question
In order to achieve the stated research aims, the following research questions is proposed:

*How do BERT-based language models compare to traditional machine learning models in NLP on the fake news detection task on human and machine-produced text?*
The subsequent sections are organized in the following manner. The Theoretical Framework section introduces the topics of news and fake news and provides their definitions adopted in this paper. Additionally, it presents the theoretical framework for studying fake news detection from the perspective of data available for analysis, a variety of models that can be employed, and an overview of some of the currently available dataset. The Related Work section describes existing literature on utilizing textual features for the fake news detection task, employing Transformer-based models on the fake news detection task, and the production of artificial fake news. Furthermore, it identifies the limits of the existing literatures and presents two contributions to the literature this paper aims to achieve. The Methodology section establishes proper research foundations from philosophical perspective, acquaints the reader with the dataset used in this paper as well as the text pre-processing and extraction techniques. Furthermore, it provides the reader with introduction to the machine learning and deep learning models employed in this paper and model optimization and evaluation methods. Moreover, production of artificial data and methods for fact-checking are introduced. The Results section presents the performance of the employed models on three classifications tasks utilizing the dataset as well as artificially generated text. The Discussion section explores the results, provides the answer to the research question, and offers perspectives on limitations, suggestions for future work, and critical reflections. Last but not least, the paper rounds off with concluding remarks in the Conclusion section.
Theoretical Framework

This section will introduce a theoretical framework used for studying fake news detection in addition to a theoretical background for news and fake news. The framework defines terms, methods and theories necessary for appropriately establishing a foundation for fake news detection.

News

News has become an inherent part of our every-day lives, nowadays people can consume and share news anywhere at any time. News is, however, a broad term that many people think of differently, and thus researchers, scholars and journalists have come up with different definitions of the term.

Cambridge Dictionary defines news as: “information or reports about recent events” (Cambridge Dictionary, 2016). Charles Anderson Dana, who worked as an editor of the Sun, considered news anything novel and interesting to a large part of a community, hence, described it as: “News is anything which interests a large part of the community and which has never been brought to their attention.” (McKane, 2014, p.1). Another, more abstract definition of news was presented by Tuchman who perceives news as “window on the world” and by looking into this window one should see what they want to know, need to know, and should know (1978, p. 1). Some researchers believe that news is tied to or even dependent on an event, according to Wilbur Schramm news is: “something perceived after the event … an attempt to reconstruct the essential framework of the event” (1949, p.259). Scholars, journalists and ordinary people possess different views and definitions of news. However, most agree that news is a description of an interesting and/or significant event. Thus, this is the definition of news that will be employed in this paper.

Humans are social beings and have an interest in events that might influence their lives. For centuries, people have established news organizations to shape and circulate knowledge (Tuchman, 1978). One of the first forms of news organizations were town criers (Park & Burgess, 1967 in Tuchman, 1978). Town criers were tasked with making public announcements and share important news with the inhabitants of a particular town. Since information sharing was rather inefficient and information from distant parts of the world was simply not available to the wide public, news usually had a local character. Therefore, news was easily verifiable, due to people’s physical proximity to an event as well as clear identity of the news’ author.
As opposed to commoners, rulers and patricians of the medieval age possessed a genuine interest in important events taking place in different parts of the then-known world. Receiving such information on a regular basis at that time could only be enabled by travelers sharing news from the parts of the world they visited (Pettegree, 2014). However, conflicting news happened to be present even at this age, which posed a challenge to rulers and powerful people because they had to decide what information to trust (Pettegree, 2014). The main indicator of the news’ credibility was the reputation of the messenger who delivered the news. This meant that news delivered by a trusted source was considered more trustworthy than news delivered by an unfamiliar source (Pettegree, 2014). By the 16th century, newsmen became more sophisticated and tended to label some messages as unconfirmed, moreover, in some cases a ruler would patiently wait for more messages confirming the occurrence of an event before making any important decision (Pettegree, 2014).

The invention of printing press in the mid-15th century offered new and more efficient ways of sharing news with the wide public. The first form of printed news accessible to ordinary people was news pamphlets (Pettegree, 2014). News included in these pamphlets usually consisted of exciting events, battles or sensations, furthermore, the authors of the pamphlets often did not hesitate to exaggerate in their texts in order to attract more readers (Pettegree, 2014).

As pamphlets with interesting and entertaining content enjoyed high popularity, newspapers containing plain and straightforward facts struggled to attract larger audiences. Despite a relatively fast spread of the newspaper, the majority of society did not have a need for receiving news about world affairs on a regular basis (Pettegree, 2014). Given that events mentioned in newspapers were not directly related to them nor were entertaining, people had little incentive to buy and read them. Therefore, teaching and convincing people to regularly consume news about events outside of their region to better understand world affairs was a necessary and tedious process. However, all these efforts ended up being successful as at the end of the eighteenth-century the newspaper became a part of everyday life and a primary source of news for many people (Pettegree, 2014).

Before the widespread adoption of the internet, physical newspapers were the main source of news distributed to ordinary people. Social media and the internet offer a unique space for anyone to create, comment and share any kind of news, and reach a mass audience (Tandoc, Lim, & Ling, 2018). Thus, even people who are not journalists can exploit all the possibilities of blogging and utilize social
media to spread a message (Tandoc et al., 2018). Sharing and consuming news from social media became a phenomenon of the 21st century and despite a high risk of disinformation/misinformation an analysis from 2018 showed that around 68% US adults get their news from social media (Shearer & Gottfried, 2018). Traditional newspapers have also moved to this virtual space in order to offer their readers information in a faster and more efficient manner.

The perceived value of news depends on its relevance or appeal to a particular individual. After all, the popularity of a newspaper is a main incentive for news creators to continue operating. Thus, it is important to realize that most newspapers are private businesses with a goal of attracting as many readers as possible, hence, make a profit (McNair, 2000). Historically, this implied selling physical newspapers whereas nowadays this focus has shifted to attracting visitors to a website. Moreover, the impact of news varies greatly. Some tabloid newspapers mostly publish news closely related to personal lives of famous people, thus their impact on the society is rather trivial as this kind of news serves primarily as a source of entertainment. On the other hand, a serious newspaper investigating and depicting important international affairs and politics can influence the opinion of the public to support a significant political or societal decision.

Given the impact that news can have, the truthfulness of news is of paramount significance. However, some people with malicious intentions try to take advantage of the power of news by creating and spreading fake news to influence and shape opinions of others.

**Fake News**

Modern technology enables people to consume news related to any event anywhere in the world. People now have access to more information than ever before, however, this ease of creating, sharing and consuming news comes at a price. The overload of information, including conflicting information, challenges people’s capability to recognize true news from fake. In this section, the phenomenon of fake news will be described in addition to five primary types of textual fake news.

Several researchers refer to fake news as news that is intentionally false in order to mislead (Allcott & Gentzkow, 2017; Klein & Wueller, 2017; Mustafaraj & Metaxas, 2017). This paper will use the definition of fake news proposed by Wardle (2017) who defines fake news as an output of misinformation (unwittingly creating and sharing false information) as well as disinformation.
(intentionally creating and sharing false information). The main reason for employing this definition, is the fact that the digitalization of news completely changed news distribution and the general perception of news (Tandoc et al., 2018). While in the past people expected news to be written and provided by journalists working in well-established and reputable newspapers, nowadays news on social media is by many perceived as credible (Shearer & Gottfried, 2018). Despite news sources often being unknown or untrustworthy, many social media users rarely verify the information they share or consume, hence, some news considered as fake are shared by people not realizing the veracity of the content (Tandoc et al., 2018).

**Typology of Fake News**

In their article, Tandoc et al describe six primary types of fake news; news satire, news parody, news fabrication, photo manipulation, advertising and public relations, and propaganda (2018). Since this paper focuses on textual news, visual fake news will not be covered in this section.

**News Satire**

News satire as a form of fake news intends to mock television news programs using humor or exaggeration to present the latest news. News satire programs mimic the style of a television news program and often focus on current events (Tandoc et al., 2018). News satire is easily recognizable as fake news by a majority of the audience who perceive it as a source of entertainment. Despite an obvious intention to entertain, this kind of fake news might be very influential. By utilizing humor or exaggeration, these programs can mock or criticize certain claims of politicians or influential people, hence, shape public opinion as well as political trust (Brewer et al., 2013 in Tandoc et al., 2018). Thus, the content of news satire comprises intentionally fake news that is based on real events (Tandoc et al., 2018).

**News Parody**

Similarly to news satire, news parody provides humorous news-like reports by mimicking traditional mainstream news media (Tandoc et al., 2018). The main difference between news parody and news satire is the use of facts. While satire uses facts to describe events in an entertaining or absurd manner, parodies are inspired by real events, however, the final product of a parody is a completely fictitious news story (Tandoc et al., 2018). The main assumption regarding the news parody is that both parties (creators and readers) are fully aware of the humor as well as the falsity of the information (Tandoc
et al., 2018). In spite of creating and publishing ridiculous and absurd news, news parody as well as news satire often highlight mistakes and faux pas of the news media, hence, serve as watchdogs to increase professionalism among journalists.

**News Fabrication**

As opposed to the two previous types of fake news, the purpose of news fabrications is to intentionally produce fake news and spread it to disinform. News Fabrication encompasses articles that are not based on facts but rather mimic the style of real news articles to gain credibility (Tandoc et al., 2018). Moreover, authors of these articles often use unverifiable facts or create an illusion of objectivity within their articles (Tandoc et al., 2018). Authors of fabricated news mimic the presentation and style of traditional media to enhance the credibility of their news, furthermore, the trustworthiness of this kind of news increases when shared by a trustworthy or respected person on social media (Tandoc et al., 2018). Fabricated news with a strong political overtone enjoys popularity mostly in societies with social tension and lack of trust in societal establishments (Tandoc et al., 2018). Furthermore, authors of fabricated news can utilize news bots that share their news on social media, which gives users an illusion that the news is read and liked by others (Tandoc et al., 2018).

**Advertising and Public Relations**

Fake news can also be exploited by public relations practitioners to create an illusion of providing real, unbiased news in order to advertise products (Tandoc et al., 2018). Thus, this kind of news is biased and often mentions only positive aspects of particular products, while still including some factual data to provide the readers with impression that they are reading real news. Clickbaits are also part of this type of fake news, as the clickbait headlines’ purpose is to use sensations or interesting captions (often without any factual support) to attract the attention of a large number of users and subsequently make them click on a certain link that usually redirects the users to some commercial website (Tandoc et al., 2018).

**Propaganda**

Propaganda constitutes news stories created by political entities with the goal of influencing public opinion (Tandoc et al., 2018). Despite being regarded as fake news, propaganda is often based on facts, however, only the facts that fit the ideology of the author are highlighted and promoted while other facts are usually slandered (Tandoc et al., 2018). Hence, propaganda shares some characteristics
with advertising as it uses real facts to describe a particular perspective in the best light in order to convince readers or viewers (Tandoc et al., 2018).

Fake news can be utilized by many individuals and institutions with different intentions. Some use fake news to merely entertain their followers or viewers while admitting the falsity of the news. On the other hand, some actors exploit fake news to benefit some entity by manipulating and shaping opinions of others. The increased occurrence of sophisticated fake news has turned confirming veracity of news into a real challenge that can only be overcome by fact-checking and successful fake news detection.

**Fake News Detection**

Having outlined different kinds of fake news and their respective purposes, we now turn our attention to describing a theoretical framework intended for fake news detection. However, it is critical to first understand why algorithmic fake news detection has received attention in the academic community as well as the general public in recent years.

As Shu et al. put it; “Fake news itself is not a new problem.” (2017, p.3). Though fake news is by no means a recent phenomenon, the digitalization of news publishing significantly amplified their reach and effect. Social media in particular, but also all the technologies they are built upon, revolutionized the dissemination of information to an unprecedented speed and scale. The increased opportunity to spread fake news to a massive number of users on social media that are now at fingers’ reach is dangerous, especially because humans’ ability to discern fake news is limited due to psychological vulnerabilities such as confirmation bias (Kumar, West, & Leskovec, 2016; Shu et al., 2017). Indeed, a meta-analysis of 206 documents found that humans on average performed just 4% better than chance at discriminating truths from lies in a study of deception judgements (Bond & DePaulo, 2006). Recent research from Stanford university revealed discomforting findings from assessing civic online reasoning skills, the ability to determine the validity of information consumed digitally, of over 7800 students, many of which are considered “digital natives”. The study found that: “More than 80% of students believed that the native advertisement, identified by the words ‘sponsored content,’ was a real news story” (Wineburg, McGrew, Breakstone, & Ortega, 2016, p. 10). Some students even mentioned that it was sponsored content but still believed that it was a news article. This suggests that many students have no idea what “sponsored content” means and that this is something that must
be explicitly taught as early as elementary school.” (Wineburg et al., 2016, p. 10). Additionally, when studying undergraduates’ reasoning skills using tweets with a political agenda, the Stanford Group found that students failed to examine the origin of tweets and the intent behind the information in them. This was due to a lack of navigational skills on social media, especially concerning tweets with a political agenda (Wineburg et al., 2016).

Fortunately, there are alternative approaches to detecting fake news rather than simply relying on an individuals’ capabilities. In fact, several studies have shown that automated classifiers outperform human judgement in discerning Fake News, emphasizing the need for algorithm-based approaches to fake news detection (Kumar et al., 2016; Myle Ott, Yejin Choi, Claire Cardie, & Jeffrey T. Hancock, 2007; Pérez-Rosas, Kleinberg, Lefevre, & Mihalcea, 2017).

**Digital Media and Fake News**

With studies showing that humans do not generally possess sufficient reasoning skills to assess the veracity of digital news and the potential of automatically detecting them employing algorithms instead, this paper presents the challenges and opportunities that the digitalization of news publishing have spawned in regard to the spread and detection of Fake News.

The digitalization of news publishing fundamentally changed how information is created, shared, and consumed. Particularly, there has been a shift from a centralized, broadcasting of information from journalists and news agencies with a majority of people acting as passive consumers, to a state where social media enables passive consumers to become co-creators of the information and disseminators by actively participating in the network and engaging with other users. Though this empowerment of users provides beneficial opportunities for useful feedback mechanisms, it creates the possibility for malicious users to exploit it. Furthermore, the pursuit of personalization for each and every user on internet platforms for the purpose of providing relevant information has been a double-edged sword as it enables what is referred to as “echo chambers” (Kumar et al., 2016). Thus, two primary concepts that have gained relevance due to the increasing proliferation of fake news on social media are: malicious actors, and echo chambers, which will be described in detail in the following sections.
Malicious Actors

Malicious actors can be both humans and computer algorithms, also referred to as ‘bots’, whose intention is to mislead a news consumer into believing a false piece of information by echoing it or supporting it directly (Kumar et al., 2016). Thus, their main purpose is to spread fake news fast and deep on digital media platforms and/or make it seem more credible (Kumar et al., 2016). The presence of such malicious actors is predicated by the fundamental design of social media, offering low costs of signing-up originally for the purpose of attracting a large number of normal users and leveraging network effects on platforms (Shu et al., 2017). These malicious actors often aim to trigger an emotional response from ordinary users on the social media platform to get them to interact with their fake news and spread it further on the platform (Shu et al., 2017). Particularly, bots can reach formidable scale. Research from the Oxford Internet Institute found that during the week before election day in the 2016 presidential election in the United States, 19 million bots tweeted supporting information for either of the candidates (The Computational Propaganda Project, 2016). Moreover, though the impact of the bots was most visible during the aforementioned presidential election, these political bots have been employed in Western Democracies such as Italy, Germany, and the UK and less democratic countries such as Russia, China, and Turkey alike (The Computational Propaganda Project, 2016).

Echo Chambers

While personalized information increases the perceived value an individual can gain from participating on a digital media platforms, it can result in a dire side-effect referred to as ‘echo chamber’ or the ‘echo chamber effect’. It is the result of self-inflicted polarization when individuals enter social groups and follow ideological pages or other individuals with whose ideas they already agree (Kumar et al., 2016). This effect is further propagated by personalization algorithms of search engines or social media platforms, which suggest content that is similar to what an individual currently consumes or content that is consumed by similar individuals, thus implicitly creating an ideologically separated chamber. Consequently, echo chambers aid in spreading fake news due to lowering the perceived need for critical fact-checking as the result of ‘social credibility’, which describes a psychological factor affecting an individual’s judgement of credibility of an information source based on the number of other individuals who consider it credible (Kumar et al., 2016; Shu et al., 2017). Furthermore, another psychological factor coined ‘frequency heuristic’ describes that the
frequency of exposure to both true as well as false information is correlated with perceived accuracy of the information (Kumar et al., 2016; Shu et al., 2017).

Although both malicious actors and echo chambers amplify the effects of fake news, they can provide useful insights into how to counter the spread of fake news. The following section contains a description of a theoretical framework for studying fake news detection, utilizing the news content as well as the additional data on malicious actors and echo chambers.

It was chosen to adapt the theoretical framework for fake news detection from Shu et al. (2017) due to it being data mining oriented and given that this paper will study fake news detection from a data perspective. Shu et al. (2017) define the fake news detection task as a binary classification problem. The authors propose the aforementioned general framework for fake news detection as comprising two phases, namely; Feature extraction and Model creation which are described in detail below (Shu et al., 2017).

Feature Extraction
The fake news detection function takes two types of inputs, also referred to as features. Shu et al. (2017) define these as News Content Features and Social Context Features, whereas the former describes features from the news content, the latter relates to auxiliary social context information.

News Content Features
News Content Features describe information regarding a news article (Shu et al., 2017). Its main attributes are the source, the creator of the news, headline, the title of the news attempting to attract the attention and sum up the main idea, body text, containing the details of the news, and image/video, usually as a part of the body providing additional visual information to the news story (Shu et al., 2017). Based on the raw data from the news, various types of features representation can be engineered to assist in identifying specifics of fake news (Shu et al., 2017). These representations can be organized into two groups, Linguistics-based and Visual-based (Shu et al., 2017). The former explores the tendency of fake news to contain writing styles that invoke emotional responses in the readers, which can be identified by features extracted on characters, words, sentences, or documents-level and often include frequency of words, average number of characters in words, usage of quotes etc. (Shu et al., 2017). The latter describes features extracted from visual material accompanying the
textual part of a piece of news and include visual as well as statistical features such as image clarity score or a number of images (Shu et al., 2017).

**Social Context Features**

Social Context Features refer to auxiliary social context information such as how news proliferates and how users engage with it on social media platforms. Similarly to News Content Features, Social Context Features are meaningfully organized into three categories; user-based, post-based, and network-based (Shu et al., 2017).

User-based features can be utilized to counter the spread of fake news, particularly those created and spread by malicious users, which are based on the interaction between the users and the news (Shu et al., 2017). Though post-based features might at first glance seem to be more related to the news content features category, they belong to the social context features category because they capture the general public’s reactions to social media posts (Shu et al., 2017).

Shu et al. (2017) differentiate networks into stance networks, co-occurrence networks, friendship networks, and diffusion networks. In order to be able to extract features from these networks, they need to be constructed first. Once successfully built, the network metrics, like a clustering coefficient, can be used as feature representations, thus network-based features (Shu et al., 2017).

Despite Shu et al. (2017) recommending the use of all of the available features for fake news detection, this paper investigates the use of News Content Features, specifically linguistic-based features extracted from the body text of news articles. Although limiting the features to just one source admittedly reduces the signal from the available data that can be useful for detecting fake news, it enables the outcomes of the analysis to be generalized to all news sources. Additionally, Undeutsch (1967) hypothesized that fabricated or fictitious stories differ noticeably from accounts of real-life events, thus the news text itself contains a trace of fictitiousness that a machine learning model could identify and use to detect fake news. This is further propagated by several studies which have investigated if fake news exhibit different textual traits from true news. One study concluded that lexical features can be used to distinguish the reliability of news sources (Rashkin, Choi, Jang, Volkova, & Choi, 2017). Other studies have also concluded that lexical and syntactic features of a body text can be utilized in the process of discerning real news from fake (Pérez-Rosas et al., 2017).
Model Creation

Understanding which features can be extracted from news is essential, however, without utilizing the features to build models that can successfully identify fake news, the features themselves would yield only limited benefits. To meaningfully organize different types of models used to detect Fake News, Shu et al. (2017) organize them into two groups based on the features that the models take in as inputs, although not exclusively only those features, namely: *News Content Models* and *Social Context Models*.

News Content Models

Models in this category rely on data from the news content and on existing factual sources to classify whether a piece of news is fake or not (Shu et al., 2017). These models are either knowledge-based or style-based, depending on the approach to the fake news detection task (Shu et al., 2017).

Knowledge-based approaches describe the most straightforward approach to assessing the veracity of a news article, by utilizing external sources to provide the context for the information contained in the news article (Shu et al., 2017). The knowledge-based approach is also referred to as ‘fact-checking’ and there has been substantial effort to automate this process (Shu et al., 2017). It further breaks down into three types of fact-checking; expert-oriented, which relies on human experts to undertake the laborious task of investigating a claim and providing a verdict regarding its credibility (websites like PolitiFact or Snopes), crowdsourcing-oriented, which leverages the ‘wisdom of the crowds’ by empowering users to label suspicious news and then have their ratings aggregated to produce the final veracity assessment (for example Fiskkit), and lastly computational-oriented, which aims to automatize and scale the fact-checking process but relies on external sources such as the open web or a structured knowledge graph for the claim validation (Shu et al., 2017).

While knowledge-based approaches depend on external knowledge, style-based approaches exploit the fact that fake news use an atypical writing style due to the intention to deceive in a believable manner (Shu et al., 2017). Style-based approaches are categorized into deception-oriented and objectivity-oriented. The former focuses on identifying the statements containing deceiving information either by utilizing probabilistic context-free grammars (PCFG) or learning the difference between deceptive and normal statements, employing rhetorical structure theory (RST) or deep learning models such as convolutional neural networks (CNN) (Shu et al., 2017). The latter is aimed
at identifying style signals that could reveal a decreasing objectivity of the news, such as hyperpartisan styles and yellow-journalism. The Hyperpartisan style is characterized by an extraordinary reaction towards a political party, while yellow-journalism contains insufficiently researched information and striking headlines due to appealing to the emotions of the reader (Shu et al., 2017).

Social Context Models

These models make use of features that stem from the social media platforms’ design, created by the users interacting with and sharing news on the platforms (Shu et al., 2017). Shu et al. point out that there have not been many approaches taking advantage of these available features, thus, they describe two main types: stance-based and propagation-based.

Stance-based approaches utilize user-generated reactions to a piece of news to infer its credibility (Shu et al., 2017). Shu et al. define the task of stance detection as: “automatically determining from a post whether the user is in favor of, neutral toward, or against some target entity, event, or idea.” (2017, p. 7). These users’ stances can then be used to assess the veracity of the piece of news (Shu et al., 2017).

Propagation-based approaches study the interrelations between relevant social media posts to infer news veracity (Shu et al., 2017). They are based on the assumption that the veracity of a news event is related to the veracity of relevant social media posts (Shu et al., 2017). These posts are studied in homogenous credibility networks, containing exactly one type of entity such as a news event, and heterogenous credibility networks, containing multiple entities such as posts and news events, (Shu et al., 2017).

As aforementioned, this paper investigates models that learn to detect fake news by learning from linguistic-based features extracted from the News Content. Therefore, this paper studies the performance of News Content, style-based models utilizing pre-trained deep learning language models as well as traditional machine learning models to learn to differentiate between deceptive and normal statements.
Datasets

Although a meaningful organization of features (data inputs) and models into a theoretical framework is crucial for standardization, effectiveness, and efficiency of academic research, to avoid issues such as duplication, what bridges theory with the real world is its application. However, several research papers recognize that collecting fake news for the purpose of dataset creation is a challenging and labor-intensive process (Kumar et al., 2016; Rubin, Chen, & Conroy, 2015). Two major challenges are the need for experts’ judgement of the veracity of news and class imbalance in data (Kumar et al., 2016; Shu et al., 2017). The former refers the meritorious work of expert journalists, fact-checkers, and crowd-sourced workers who gather auxiliary data, analyze the news context, and verify the credibility of the news article (Shu et al., 2017). The latter describes inherent underrepresentation of fake news, pointing out that it accounts for less than 10% of all news (Kumar et al., 2016).

Fortunately, researchers have made multiple public datasets available. The following is an overview commonly used dataset for fake news detection tasks.

- **BuzzFeedNews**\(^1\) covering news on Facebook in a week close to the 2016 US Presidential election, with each claim having been fact-checked by five BuzzFeed journalists (Shu et al., 2017).

- **LIAR**\(^2\) a dataset consisting of 12,836 short statements, sampled from a variety of contexts such as TV ads, campaign speeches, etc., recovered from PolitiFact.com’s API, and annotated by its editors with one of six veracity labels: pants-fire, false, barely-true, half-true, mostly-true, and true (W. Y. Wang, 2017).

- **BS Detector**\(^3\) is a dataset that was collected by and named after a browser extension having scraped 244 websites labelled as “bullshit” because of containing links referring to untrustworthy external sources, which were identified on a manually curated list of domains.

- **CREDBANK**\(^4\) is a large-scale dataset containing over 60 million tweets collected between 10-Oct-2014 to 26-Feb-2015, related to 1049 real-world events, and labelled by 30 Amazon Mechanical Turkers on a scale from -2 to +2 (‘-2 Certainly Inaccurate’ to ‘+2 Certainly Accurate’) (Mitra & Gilbert, 2015).

---

2. https://github.com/thiagorainmaker77/liar_dataset
• *FakeNewsNet* is a data repository consisting of two datasets containing news content, social context and dynamic features (spatial-temporal) with news collected from reliable fact-checking websites PolitiFact and GossipCop, which was motivated by the lack of datasets for studying fake news detection that contained all three types of features (Shu, Mahudeswaran, Wang, Lee, & Liu, 2018).

• *MultiFC* is a dataset composed of 34,918 labeled claims in English with rich meta-data, such as the speaker of the claim, URL to the claim and the date of the claim, collected from 26 different fact-checking websites (Augenstein et al., 2019). *MultiFC* stands out from the rest of the currently available datasets due to its size as well as its quality of being a dataset of naturally occurring claims that were assigned a veracity label by expert journalists (Augenstein et al., 2019).

Having multiple datasets available for testing models developed for the task of automatically detecting fake news is a positive sign of a thriving academic environment, particularly because a lack of sufficient examples of fake news has historically been a major obstacle for previous research (Rubin et al., 2015).

However, the fight against fake news is being fought on several academic fronts. Among the aforementioned are the analysis of humans’ ability to detect Fake News, studying the economic impact and motivation behind fake news, developing theoretical frameworks for studying fake news detection, and creating and making datasets available for other researchers to develop and test models. In addition, among those previously unmentioned belong empowering users on social media platforms to fact-check news (Vo & Lee, 2018), setting-up fake news detection competitions (WSDM on Kaggle (Kaggle, 2018) and Fake News Challenge (Fake News Challenge, 2016)), and last but not least the testing of models for detection and generation of fake news, which is the focus of this paper, and thus is described in detail in the following section.
Related Work

This section will establish an overview of existing literature focusing on fake news detection and fake news generation using language models. In addition to providing an overview of existing literature on the topics, this section will also outline how this paper will contribute to the existing research field.

Despite the field of fake news detection having gained increased attention from researchers in recent years, many researchers agree that few datasets of appropriate size and quality are publicly available which has limited the extent of existing research (Ahmed, Traore, & Saad, 2017; Augenstein et al., 2019). Despite this limitation, research has been conducted using a variety of machine learning and deep learning models on different datasets.

Text Features From Claims

One area of fake news detection research is focused on detecting the veracity of a statement by solely analyzing features extracted from the text of a claim itself.

One such study applied 23 different supervised machine learning models on three different datasets all consisting of a claim of text and a veracity label. The datasets varied in size with the smallest one containing 75 instances and the biggest one containing 44,898 instances. Following a series of pre-processing steps including tokenization, stemming and stop-word removal, a set of features were extracted for each dataset using term frequencies. For each dataset, a distinct minimum count threshold for each feature was specified. The models were subsequently applied to a binary classification task on all three datasets. While results varied amongst the datasets, the study found that decision trees generally performed best with accuracy scores ranging between 74 – 96%, depending on the dataset (Ozbay & Alatas, 2020).

A similar study created a new dataset by combining real news data from Reuters.com with fake news data collected from a Kaggle dataset that focused on claims related to the 2016 US presidential election. Simple pre-processing such as removing stop words, lower casing text, removing punctuation and stemming were performed. Using both Term Frequency and TF-IDF with N-gram values ranging from [1,4] and different thresholds for the number of features ranging from 1000 to 50,000; the study found that linear classifiers such as Logistic Regression, Stochastic Gradient Descent (SGD) and Linear SVM performed better than non-linear classifiers. The highest score was
achieved using a linear SVM model on unigram TF-IDF features and feature size of 50,000. The study also experimented with other models such as K-Nearest Neighbors and Decision Trees, although these performed worse than the linear models (Ahmed et al., 2017).

Ensemble methods have also been used to investigate which combinations of models and features achieve the best performance in discriminating real against fake news.

One such study concatenated two different datasets in order to create one unified dataset that could be used for fake news detection. The study used the FakeNewsNet dataset which consists of fact-checked statements labelled from Politifact compiled by researchers at Arizona State University. In addition, the McIntire Dataset that contains claims from the 2016 US Presidential election was used. The two datasets were concatenated into a single dataset with a text of claim and label column. The final dataset consisted of 5,405 training instances and 1,352 test instances with a primary topic of the data being US politics. The training dataset was evenly balanced between the real and fake label. The data was tokenized, and words transformed into TF-IDF and word embedding representations. In addition, stemming and lemmatization were performed which resulted in some words being omitted from the final dataset to prevent performance issues. The top 25,000 features were subsequently selected. The paper outlines three distinct feature sets that contain distinct stylometric features of the text of claim that are useful for discriminating between real and fake news. By conducting several experiments, the paper found that only some of these features provided additional value to the classification task. The paper applied a wide array of different classifications methods: Random Forest, Naive Bayes, SVM, KNN, Logistic Regression, Adaboost and Stochastic Gradient Boosting. Using ensemble methods such as bagging and boosting, the paper investigated which combinations of word embedding and stylometric features provide the highest accuracy scores. The paper concludes that combining word embedding vector representations with stylometric features such as number of quotes and number of uppercase letters provide the best model performance. Using a gradient boosting method, the paper achieved an accuracy score of 95.49%. The paper concludes that using features solely related to the body text of a claim can be useful in discriminating real against fake news (Reddy, Raj, Gala, & Basava, 2020).
Fake News Detection Using Transformer Models

The Transformer architecture (Vaswani et al., 2017) is utilized by several state-of-the-art language models within the field of NLP. Due to the novelty of the Transformer architecture, a limited number of research papers have applied models based on the architecture to different varieties of the fake news detection task.

One paper utilized BERT to conduct a classification task on different news articles. By using articles labeled as “bullshit” by users of a browser extension as fake news and real news from sources like The New York Times, the paper used BERT, CNNs and LSTMs to solve a binary classification task of labeling articles as “true” or “fake”. The paper finds that all three models outperform previous approaches to the same task and concludes that neural networks trained solely on text features should be utilizable for the task of fake news detection (Rodríguez & Lloret Iglesias, 2019).

A different such study investigated the identification of propaganda based on textual features using models such as BERT and RoBERTa. Using an imbalanced dataset of claims with binary labels the task was to determine if a claim was propaganda or not. The paper achieved an F1 score of 0.43, indicating that there is significant room for improvement (Aggarwal & Sadana, 2019).

Another study focused on the detection of fake news on Twitter and Weibo using both textual and visual features. The paper created a new model that used BERT to process text features combined with a CNN to extract visual features from images associated with each post. The paper used a dataset from Twitter that was verified by cross-checking other sources (Boididou et al., 2016) in addition to a dataset collected from Chinese state-controlled media Weibo. By combining text features with visual features extracted from images, the paper achieved better results than some EANN networks proposed by other papers (Singhal, Shah, Chakraborty, Kumaraguru, & Satoh, 2019).

Other studies have focused on applying BERT to classifications tasks related to irony detection (C. Zhang & Abdul-Mageed, 2019) or using BERT and similar Transformer-based models to establish contextual word embeddings to be used by non-Transformer models (Autef, Matton, & Romain, 2019; Pham, 2019).
Stance detection using Transformer models is also a task that has gained attention. For the stance detection tasks, a news headline is provided. The goal is to choose between $n$ statements each associated with a label that denotes its agreeability with the original headline. Models such as BERT, RoBERTa and XLNet have been applied to these tasks and achieved promising results (Dulhanty, Deglint, Daya, & Wong, 2019; S. Liu, Liu, & Ren, 2019; Slovikovskaya, 2019), and new models have been developed by modifying existing Transformer-based models to accommodate these tasks (Jwa, Oh, Park, Kang, & Lim, 2019).

Artificial Fake News

Although language models have seen much progress in recent years, research on artificially produced fake news is sparse. Given that generative language models that are capable of producing human-like text are relatively new, research on how these models are capable of performing a certain task like fake news creation and its influence is limited. Because of this, most of existing research focused on fake news is centered around human-produced fake news (Waweru Muigai, 2019; Westerlund, 2019).

Despite the overwhelming majority of fake news research focusing on human-made fake news, one study developed a model capable of producing artificial fake news. The study is centered around a security approach with the aim of investigating potential threats and solutions to artificially produced fake news. The paper introduces the Grover model (Zellers et al., 2019). Grover is a generative language model based on the GPT-2 architecture. The model is trained on 120GB of text data from 5,000 different news domains. The model is capable of producing artificial news articles with features such as date of publication, authors, publication source in addition to a full body of text. The paper introduces the term “neural fake news” which denotes fake news text produced by a model based on neural networks. By conducting a comparative study between articles produced by Grover and articles produced by humans, the paper finds that propaganda articles produced by Grover were deemed more plausible to be authentic than propaganda articles written by humans. The articles were judged on their content, trustworthiness and the overall writing style (Zellers et al., 2019). Despite this, the paper also found that non-propaganda articles produced by Grover are of lower quality than their human counterparts. The paper concludes that artificial fake news poses a potential threat for society and argues that making text generation models publicly available is a potential method for mitigating future risks.
Limitations of Existing Research

This section will outline the limitations of existing literature and how this paper contributes to the existing research field.

While existing research has applied Transformer-based models to tasks within the domain of fake news detection, this research has often utilized small datasets (Hiramath & Deshpande, 2019), focused on specific types of fake news or performed tasks that are inherently different from a binary classification task. The datasets used for fake news detection are often limited in variety and size due to the claims often being sourced from a small number of domains which leads to datasets that usually revolve around a limited selection of topics and entities. On top of this, the fact-checking methods applied by the data sources differ greatly. Most existing research does not utilize datasets with claims fact-checked by experts but often rely on claims with crowd-sourced veracity labels. This results in data that is limited in variety, size and quality.

While several studies have investigated the use of external features to predict the veracity label of a claim and have achieved better scores than solely using text-based features (Augenstein et al., 2019; Shu et al., 2018), these studies have not investigated the capabilities and performance of the Transformer-based models solely on the text of the claims. This paper will therefore disregard external features and focus solely on the text of the claims to determine if models based on the Transformer-architecture are capable of improved detection of patterns than previous models.

Much of the existing Transformer-based fake news detection research is focused on stance detection tasks and not on binary classification tasks on expert fact-checked data. In addition to this, much of the existing research revolves around a specific type of fake news. Lastly, most existing research on Transformer-based models is usually restricted to using a single model, thereby disregarding the opportunity for performance comparison amongst different models based on similar architectures.

Given that generative language models that are capable of producing human-like text are relatively new, little research has gone into investigating how these models can be used for fake news generation or how the produced text can affect the performance of models tasked with detecting fake news. Furthermore, existing research on fake news generation has focused mainly on producing propaganda articles which constitutes just one type of fake news (Zellers et al., 2019).
Based on these shortcomings in existing research, this paper will contribute to the research field in the following ways:

1. This paper will apply three BERT-based language models based on the Transformer-architecture to a binary classification task to the, at time of writing, largest publicly available dataset of expert fact-checked naturally occurring claims sourced from several domains constituting several types of fake news.

2. This paper will employ the GTP-2 model to produce artificial claims based on human-made claims to determine change in classification performance when training data is increased with artificially produced text.
Methodology

This section will establish the methodical foundation for this paper. This includes presenting the chosen research philosophy, the primary data, text pre-processing techniques, the applied models, fact-checking, evaluation metrics, model optimization and evaluation in addition to creation of artificial text data.

Research Philosophy

Research philosophy is a term used to describe the nature and development of knowledge (Saunders, Lewis, & Thornhill, 2009). Be it intentionally or inadvertently, researches make assumptions (Saunders et al., 2009). These assumptions about the nature of reality and human knowledge affect how researchers approach their research, the methods they choose and how they present their findings (Crotty, 1998). Research philosophy describes the assumptions researchers make about the world in which they conduct research (Saunders et al., 2009). Becoming aware of these assumptions and understanding their philosophical aspect, the researchers can examine their philosophical choices and reflect on them in respect to alternative approaches (Saunders et al., 2009). There are two major ways of thinking about research philosophy, ontology and epistemology (Saunders et al., 2009).

Ontology

Ontology describes the nature of reality in which knowledge is created (Saunders et al., 2009). Ontology comprises assumptions about how the world operates (Saunders et al., 2009). There are two aspects of ontology, objectivism and subjectivism (Saunders et al., 2009). Whereas the former represents the belief that social entities are independent from social actors, the latter stands for a position that views social phenomena as created by actions of social actors (Saunders et al., 2009). Furthermore, since a subjective aspect relies on social interaction between actors, social phenomena are being continuously revised, meaning that in order to understand a situation, it is crucial to understand its details as different interpretations can be reached (Saunders et al., 2009).

Epistemology

Whereas ontology is concerned with the nature of reality, epistemology is concerned with the creation of acceptable knowledge (Saunders et al., 2009). As realism was selected to be the most appropriate epistemological stance, a brief description of its main points ensues.
Realism is a philosophical stance that employs scientific enquiry to develop or create knowledge (Saunders et al., 2009). From an ontological perspective, realism describes reality as being experienced via senses yet having an independent existence from the mind of a researcher (Saunders et al., 2009). There are two branches of realism: direct realism and critical realism. The former describes a philosophical assumption that reality can be accurately experience via the human senses (Saunders et al., 2009). While the latter contends that experiences of reality through senses are a mere sensations of reality, though not a reality itself as senses can deceive us (Saunders et al., 2009). Furthermore, critical realists acknowledge that despite reality existing independently of human thought, our knowledge of it depends on social conditioning (Saunders et al., 2009).

Methodological Choice

All research is based on a methodological choice, mono method – quantitative or qualitative, or multi methods, and identification of the nature of the research (Saunders et al., 2009). Quantitative research employs analysis of numerical data and is commonly associated with positivist philosophical stance, deductive approach, and characterized by statistical analysis of relationships between variables (Saunders et al., 2009).

Opposite to this, qualitative research focuses on analysis of non-numeric data (such as text, images, video, etc.) and is often associated with interpretivist philosophical stance and inductive approach, although some qualitative studies start off with deductive approach (Saunders et al., 2009).

Multiple methods research design is organized into two categories, multimethod that utilizes multiple methodologies of either quantitative or qualitative design, and mixed method research that combines the two (Saunders et al., 2009). In mixed method research, qualitative data can be numerically encoded or quantitative data can be turned into text (Saunders et al., 2009). Multiple methods research is generally associated with the philosophical stance of realism, particularly critical realism, utilizing inductive, deductive, or both research designs, and is characterized by richer data collection, analysis, and interpretation (Saunders et al., 2009).

In relation to the research design, it is crucial to recognize the nature of the research. An exploratory study is the first step towards knowledge creation in a research topic without previous work (Saunders et al., 2009). Descriptive studies aim to capture the specifics of a phenomenon, though in itself might
be insufficient and therefore is often combined with the last type, the explanatory study (Saunders et al., 2009). The goal of the explanatory study is to create knowledge of the relationship between variables, typically striving to establish a causal relationship (Saunders et al., 2009).

This paper adopts the philosophical stance of critical realism, which contends that objects are independent of the human mind, as the claims that are evaluated to be fake or not exist in reality external to the minds of the fact-checkers. However, critical realism also states that knowledge of reality is dependent on social conditioning, which is relevant given that the analyzed claims in this paper had been fact-checked by humans who have been subject to social influences. Critical realists employ scientific enquiry, which is associated with objectivist ontology, reflecting the assumption that objects are independent of the human mind. For the research topic of fake news detection, it is appropriate to adopt the objectivist ontology as claims need to be verifiably fake and thus there needs to exist evidence, or lack thereof, external to the mind of the fact-checker. As this paper leverages and builds on a wealth of previous theoretical knowledge in the topic of fake news detection, it adopts the deductive research approach, which describes a research approach of testing existing theory on collected data and verifying or falsifying a theory. To accommodate the theoretical specifications, which define fake news detection as a binary classification task, the labels in the primary data needed to be reduced to binary values. Both the label reduction and the consequent analysis were conducted in a structured manner to ensure reproducibility. Moreover, choosing the largest dataset of naturally occurring claims from a variety of domains as the primary data further improves generalization of this paper’s results. As natural claims in textual form are inherently qualitative data, yet machine learning models require numerical data as input, the textual data was converted utilizing a multitude of text feature extraction methods described below. Lastly, as outlined in the related works section, to the author’s knowledge, there is a gap in academic knowledge on the utilization of models based on the Transformer architecture for fake news detection, which this paper aims to partially fill, thus, the nature of this paper’s research is exploratory.
Data

Primary Data

The primary dataset used for the fake news classification task is the MultiFC claim verification dataset released by Augenstein et al. (2019) in connection with their paper “MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims”. The dataset consists of 34,918 claims scraped from 26 different English fact-checking websites (Augenstein et al., 2019). Common for all claims is that they are non-artificial i.e. naturally occurring. In addition, all the claims have been fact-checked by professional fact-checking journalists using standards required by their source website.

The dataset was established from a wide variety of domains focusing on topics such as African news, US politics, gossip, advertisement hoaxes, local news, climate change and more. Due to the claims being gathered from different websites utilizing different terminologies and methods for conducting the fact-checking process, the dataset contains 117 unique labels which denote the veracity of the claims. The dataset is publicly available and can be downloaded as part of an ongoing competition to improve the results achieved by the paper using the same data (Lucasschaves, 2019). The dataset is provided as three tab-separated (.tsv) files. The files were loaded into a Pandas DataFrame. All subsequent data pre-processing was conducted in Python using the NLTK and Pandas libraries (Bird, Loper, & Klein, 2009; McKinney, 2010). Upon inspecting the dataset, it was discovered that it contained 34,842 claims from a total of 27 unique fact-checking domains. General information relating to the files is presented in Table 1.

<table>
<thead>
<tr>
<th>Filename</th>
<th>Instances</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev</td>
<td>3484</td>
<td>13</td>
</tr>
<tr>
<td>Train</td>
<td>27871</td>
<td>13</td>
</tr>
<tr>
<td>Test</td>
<td>3487</td>
<td>12</td>
</tr>
</tbody>
</table>

*Table 1 - Dataset Dimensions*

In addition to the claim itself the files contain different kinds of metadata related to each claim. An overview of all features can be found in Table 2.
Data Preparation and Pre-processing

Given that this paper aims to investigate the performance of traditional machine learning models and deep learning models on a binary classification task on fake news data, many pre-processing steps had to be taken to ensure that the data would be fit for the task at hand. The first step was to discard the Test file, as it did not contain a “Label” column which is necessary for conducting a supervised binary classification task with the models chosen for this paper. The Dev and Train files were concatenated along the rows to create a single file with 31,355 instances and 13 features.
Label Reduction

The original dataset contained 117 unique labels. An overview of the domains with the highest number of labels can be seen in Figure 1.

Figure 1 - Top 10 websites with the most labels

Figure 2 shows a distribution of the 20 most frequently occurring unique labels.

Figure 2 – The 20 most frequently occurring labels in the dataset
From Figure 2, it can be argued that the distribution is highly skewed. Five labels (false, mostly true, mostly false, true, half-true) represent over 50% of all instances in the dataset. Opposite to this, 60 of the 117 total labels are associated with less than 30 individual claims each. This is a result of the websites not utilizing a standardized rating scheme. Due to the high degree of unevenness in the labels it was deemed inappropriate to conduct a multinomial classification task as a majority of the labels would be underrepresented. Instead, it was decided that the label space should be reduced to binary values (true or fake) as outlined in the theoretical framework (Shu et al., 2017). In order to perform this reduction, a qualitative process of determining an appropriate label for each claim was initiated. To determine if a label should be reclassified as true or fake, the source website(s) for each claim were visited. Each unique website was visited with the purpose of finding a rating scheme that provided information as to how each claim was rated and to give more context to the rating itself. Of the 27 unique websites, 6 did not contain a fact-checking rating scheme. If a rating scheme could be found, a qualitative assessment was made to re-classify each label as true or fake. For websites where a standardized rating scheme was not available or did not exist, the assessment was based on other claims with the same label or the wording of the label itself. The 117 unique labels were reduced to 3 distinct labels: true, fake & ambiguous. The ambiguous category was added to accompany claims that were labelled in manners that did not provide any or unclear information regarding the veracity of the claim. A preview of the final label-classification transformation can be seen in Table 3. The full table can be found in Appendix 1 – Label reduction of MultiFC Dataset.

<table>
<thead>
<tr>
<th>Original Label</th>
<th>Reclassified as</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>determination: misleading</td>
<td>Fake</td>
<td><a href="http://www.voiceofsandiego.org">www.voiceofsandiego.org</a></td>
</tr>
<tr>
<td>misleading recommendations</td>
<td>Fake</td>
<td><a href="http://www.hoax-slayer.com">www.hoax-slayer.com</a></td>
</tr>
<tr>
<td>scam!</td>
<td>Fake</td>
<td><a href="http://www.truthorfiction.com">www.truthorfiction.com</a></td>
</tr>
<tr>
<td>fiction! &amp; satire!</td>
<td>Fake</td>
<td><a href="http://www.truthorfiction.com">www.truthorfiction.com</a></td>
</tr>
<tr>
<td>misleading!</td>
<td>Ambiguous</td>
<td><a href="http://www.truthorfiction.com">www.truthorfiction.com</a></td>
</tr>
<tr>
<td>inaccurate attribution</td>
<td>Ambiguous</td>
<td><a href="http://www.truthorfiction.com">www.truthorfiction.com</a></td>
</tr>
<tr>
<td>a lot of baloney</td>
<td>Fake</td>
<td><a href="http://www.huffingtonpost.ca">www.huffingtonpost.ca</a></td>
</tr>
<tr>
<td>outdated</td>
<td>True</td>
<td><a href="http://www.snopes.com">www.snopes.com</a></td>
</tr>
<tr>
<td>10</td>
<td>True</td>
<td><a href="http://www.gossipcop.com">www.gossipcop.com</a></td>
</tr>
<tr>
<td>truth! &amp; unproven!</td>
<td>True</td>
<td><a href="http://www.truthorfiction.com">www.truthorfiction.com</a></td>
</tr>
<tr>
<td>truth! &amp; disputed!</td>
<td>True</td>
<td><a href="http://www.truthorfiction.com">www.truthorfiction.com</a></td>
</tr>
</tbody>
</table>

Table 3 – Example of label transformation process
In order to accommodate data type requirements for the models, an additional column named “Label Numeric” was added. This column contained a numeric representation of each of the three new labels. Once all the original labels had been mapped to their new values, the dataset had the following dimensions:

<table>
<thead>
<tr>
<th>Label</th>
<th>Label Numeric</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake</td>
<td>0</td>
<td>16628</td>
</tr>
<tr>
<td>True</td>
<td>1</td>
<td>10629</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>2</td>
<td>4098</td>
</tr>
</tbody>
</table>

*Table 4 – Count of claims associated with each Label following label reduction*

All claims with the Ambiguous labels were subsequently dropped from the dataset. This resulted in a new dataset with the following dimensions:

<table>
<thead>
<tr>
<th>Label</th>
<th>Label Numeric</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake</td>
<td>0</td>
<td>16628</td>
</tr>
<tr>
<td>True</td>
<td>1</td>
<td>10629</td>
</tr>
<tr>
<td>Total</td>
<td>N/A</td>
<td>27257</td>
</tr>
</tbody>
</table>

*Table 5 – Dataset Label Counts following removal of claims labeled as ambiguous*

As a result of the label-reduction process, the “Fake” label was overrepresented with 5,999 more instances than the “True” label. In order to prevent biased results as a consequence of class-imbalance (Bishop, Christopher, 2006; Jacobusse & Veenman, 2016), sampling was performed to select a random subset of fake claims equaling the total number of true claims. Thus, the dataset was balanced with 10629 instances of each label class. To ensure that any model applied to the dataset would choose a random subset of the data in a non-biased manner, a sampling function was applied to the balanced dataset to ensure that all the data would be shuffled before being passed to any model. From here on, the label-reduced and balanced version of the MultiFC dataset will be referred to as the “natural claims dataset”.
**Text Pre-processing**

Text pre-processing, also referred to as text normalization, is traditionally the first step in NLP tasks (Jurafsky & Martin, 2018). Selecting appropriate pre-processing techniques, with consideration to the domain and language of the textual data, can yield substantial performance improvements in text classification tasks (Song, Liu, & Yang, 2005; Uysal & Gunal, 2014). Therefore, text pre-processing is an essential step in successfully classifying fake news, in order to avoid the colloquial ‘garbage in garbage out’ scenario, which captures the intuition that regardless of the complexity of the applied algorithm, poor quality data will inevitably yield poor quality results (Song et al., 2005; Uysal & Gunal, 2014). Text pre-processing generally consists of three components: tokenization, normalization and noise removal.

**Tokenization**

Tokenization is the process of extracting words from a running text (Jurafsky & Martin, 2018). Although extracting words might seems like a straightforward task, issues occur particularly with words containing non-alphanumeric characters such as hyphens (e.g. long-term), dots (e.g. Ph.D.), or apostrophes (e.g. didn’t) and other special characters (Jurafsky & Martin, 2018). Processed text often contains quantities or prices (e.g. 555,500.50 and $45.55) that should be extracted as one token (Jurafsky & Martin, 2018). Additionally, thousands and decimal separators differ between English and e.g. European languages, whereas in English thousands are separated with a comma and the decimal with a dot (e.g. 555,500.50), in many European languages thousands are separated with a space and decimal with a comma (e.g. 555 500,50), which further emphasizes the importance of adapting processing according to the domain and the language of the processed text (Jurafsky & Martin, 2018).

**Normalization**

Text normalization consists of techniques to transform words/tokens into their standard form and to represent words/token in multiple forms with a single form (e.g. representing ‘goooood’ ‘and ‘gud’ with good) (Jurafsky & Martin, 2018). Furthermore, text normalization can help with sparsity issues, particularly with smaller datasets where there is not a sufficient number of occurrences for some words, as text normalization would map these rarely occurring words onto a common representative form. Moreover, reducing the number of features generally speeds up additional pre-processing steps as well as the training of a classifier.
Case Folding

Case folding, also referred to as lowercasing or lowercase conversion, is a commonly applied normalization method that transforms all letters in a corpus to lower case, which aids in generalization and maintaining consistency between words/tokens. Skipping case folding might be desirable for specific text classification tasks, such as sentiment analysis, as the case can carry important information about the sentiment that would otherwise be lost at the expense of generalization (Jurafsky & Martin, 2018). However, other experiments have found that lowercasing improves the performance of classifiers, in terms of both accuracy and dimension reduction, irrespective of domain and language (Uysal & Gunal, 2014).

Stemming

Stemming can be thought of as the little brother to Lemmatization. Whereas Lemmatization converts the word to its basic meaning-bearing form, Stemming is a heuristic-based approach that revolves around removing the affixes of words (Jurafsky & Martin, 2018). Therefore, Stemming is less complex, and thus faster than Lemmatization, albeit less accurate as it sometimes returns stems that are not actual words as is demonstrated in Figure 3. Nevertheless, it may be sufficient in text processing tasks such as information retrieval, or in cases where performance is of greater importance than accuracy. The most used, and widely known stemmer is the Porter Stemmer (Porter, 1980).

Lemmatization

Lemmatization is a text normalization technique that reduces words to their root form, also referred to as their ‘dictionary’ form, called lemma. For example, words ‘am’, ‘are’, and ‘is’ have the same lemma ‘be’. Lemmatization stems from morphology, the study of how words are composed of smaller, meaning-bearing parts, called morphemes (Jurafsky & Martin, 2018). There are two major categories of morphemes: stems, the main part that provides the meaning of the word, and affixes, which contribute additional meaning (Jurafsky & Martin, 2018). The most complex lemmatization algorithms leverage complete morphological parsing of the words (Jurafsky & Martin, 2018). An example of stemming and lemmatization can be seen in Figure 3.

Word: troubling
Porter's stemmer: troub
Lemmatization: troubling

Figure 3 – Example of Porter’s stemmer and Lemmatization
Other Normalization Techniques

The text normalization techniques described in this paragraph are generally recognized by practitioners of NLP, although not extensively studied in the academic sphere.

Firstly, expanding contractions, words that are created by shortening and combining two words (e.g. can + not = can’t, I + would = I’d, etc.), might be desirable to enforce normalization of text, particularly in text classification tasks such as sentiment analysis, where the negative part (‘not’) of a contraction carries important information and can otherwise be disregarded. Secondly, converting accented characters is another step towards text normalization focusing on removing accents from translingual words such as frappé, naïve, or résumé.

Noise Removal

Noise removal is the most domain-specific part of text pre-processing. As its name indicates, it consists of steps that remove parts of the text that would otherwise act as ‘noise’ for a text classifier, and thus reduce its accuracy (Ganesan, 2019). The most common technique of noise removal omits stop words, which are words that frequently appear in text yet bear little to no semantic significance (e.g. “the”, “and”, “a/an”, etc.). Furthermore, missing values in a dataset as well as duplicated textual data entries might provide an insight to an erroneous data gathering process, however, from the perspective of optimizing the performance of text classifiers, these should be removed. Additionally, all non-Alphabetic or non-Alphanumeric characters can be removed.

These pre-processing techniques were applied to the raw textual data of the claims in the following order. Firstly, the text of the claims was lowercased, and the duplicates were removed. Additionally, contractions were expanded, and all non-Alphabetical characters were omitted. Subsequently, accents were removed as well as stop words. Lastly, the text of claims was lemmatized. After the techniques were applied, the claims were ready for extracting text features, which are introduced in the following section.

Text Feature Extraction Techniques

Although pre-processing raw text data is an essential step in NLP tasks, preprocessed text needs to be transformed into features that machine learning models can learn from. The techniques that
transform text into features are referred to as Text Feature Extraction techniques. These features can, in simple terms, be thought of as numerical representation of the statistical properties of text segments (words/tokens). In the following sections three commonly used Text Feature Extraction techniques are presented: N-grams, Bag-of-Words, and Term frequency-inverse document frequency.

N-Grams

N-gram is a sequence of $n$ contiguous items extracted from running text. The extracted items are typically words but can also be individual characters. $n$ refers to the number of extracted items in the sequence with their corresponding names using Latin prefixes, Unigram, Bigram, Trigram. An example of several word n-grams is shown in Table 6.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>“This is a running text.”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigrams</td>
<td>[‘This’, ‘is’, ‘a’, ‘running’, ‘text’]</td>
</tr>
<tr>
<td>Bigrams</td>
<td>[‘This is’, ‘is a’, ‘a running’, ‘running text’]</td>
</tr>
<tr>
<td>Trigrams</td>
<td>[‘This is a’, ‘is a running’, ‘a running text’]</td>
</tr>
</tbody>
</table>

Table 6 – Examples of word N-Grams

N-grams are sometimes also used to refer to language models that compute the probability of the next word given a sequence of words, calculated as a conditional probability computation (Jurafsky & Martin, 2018). To simplify the calculation, an independence assumption, called Markov assumption, is made so that the probability of the next word depends only on the last $n − 1$ words, which approximates the probability (Jurafsky & Martin, 2018). Thus, a Bigram would approximate the probability of the next word based on the previous word and Trigram on the two previous words. However, natural language contains long-distance dependencies that N-grams cannot capture, which researchers were already aware of in 1998 when N-gram were the, de facto, state-of-the-art NLP technique for text feature extraction (Brill, Florian, Henderson, & Mangu, 1998). Nonetheless, N-grams models have been widely adopted in NLP as well as other scientific fields due to their simplicity and scalability.

Bag-of-Words

Bag-of-Words (BoW) is one of the simplest text feature extraction techniques, as the only features it extracts are the unordered frequencies of words occurring in a document (e.g. a sentence, or a
paragraph), while disregarding their position in the text (Jurafsky & Martin, 2018). Since it extracts features per document, it is often used in document classification tasks. BoW can also be thought of as a special case of N-grams with $n = 1$, where each word from the text has the probability of being the next word equal to the frequency of the word divided by the total count of words. Although this technique is crude in its approach, it historically found several successful applications, for example in email filtering.

In this paper, the BoW representations were calculated using the CountVectorizer (CV) functionality from Sci-Kit Learn (Pedregosa et al., 2011).

**Term Frequency – Inverse Document Frequency**

Although the BoW technique has its use cases, extracting just the frequencies of words is generally insufficient for capturing the complexities of relationships between words (Jurafsky & Martin, 2018). Specifically, common words such as ‘the’, ‘and’, or ‘to’ have high frequencies, yet appearing in all documents strips them of discriminatory power, and thus renders them almost useless for document classification (Jurafsky & Martin, 2018). Therefore, it is desirable to normalize frequencies to ensure that words that are common across all documents are given low weights, whereas words that appear only in a few documents are given higher weights and thus provide improved discriminating information to the models (Jurafsky & Martin, 2018). This is what the Term Frequency – Inverse Document Frequency (TF-IDF) technique accomplishes.

TF-IDF consists of two parts, TF stands for *term frequency* and represents the raw counts of words in a document, and IDF is an abbreviation for *inverse document frequency*. The intuition behind using inverse document frequency comes from understanding that words that appear in a few documents are useful for discrimination between documents (e.g. in discriminating between spam e-mails and non-spam e-mails the word Viagra is useful for discrimination, as it rarely appears in non-spam e-mails) (Jurafsky & Martin, 2018). Therefore, using the inverse of the document frequency measure, the number of documents a specific word appears in indicates its usefulness for discrimination between the documents. Thus, multiplying *term frequency* with *inverse document frequency* creates numerical feature representations for words in a document based on all the words in the collection of documents, which are then passed as input to the text classifiers. The equation for TF-IDF can be seen in Equation 1 (Jurafsky & Martin, 2018).
\[ w_{i,j} = TF_{i,j} \ast \log\left( \frac{N}{DF_i} \right) \]

*Equation 1 – TF-IDF Formula*

Introduction to Applied Models

The following section will describe the models used to evaluate the dataset used in the paper. The section is split into two parts: Baseline Models and *BERT-Based Models*. The Baseline section introduces the models that will be used as a benchmark to compare results with the more recently developed models covered in the BERT-Based Models section. In this paper, “baseline models” refers to the traditional machine learning models. “BERT-based models” refers to the deep learning models.

Baseline Models

The following section will describe the three baseline models used for the classification tasks: Support-Vector Machines, Logistic Regression and Naïve Bayes Classifier. All three models are easy to implement, fast and have proven to provide comparable results to more sophisticated models despite their relatively simple structures and naïve assumptions. The models are therefore considered baselines for text classification tasks (Hastie, Tibshirani, & Friedman, 2009; Rennie, Shih, Teevan, & Karger, 2003; Vanderplas, 2017). All three models were implemented in Python using the SciKit-Learn library (Pedregosa et al., 2011).

Support-Vector Machines

Support-Vector Machines (SVM) is a supervised model commonly used for binary and multiclass classification tasks. The approach was developed in the 1990s and has since become popular within the field of machine learning due to its versatility and generally high performance (Cortes & Vapnik, 1995; Géron, 2017; James, Witten, Hastie, & Tibshirani, 2013). SVMs solve a classification task by establishing a linearly separating hyperplane, referred to as a “decision boundary”, between the data classes. SVMs find the optimal hyperplane by searching for the maximum marginal hyperplane i.e. the hyperplane that establishes the largest possible margin between the hyperplane and the data instances belonging to their respective classes (Géron, 2017; Han, Kamber, & Pei, 2012; James et al., 2013). The margins are defined by two hyperplanes, \( H1 \) and \( H2 \) which are both parallel to the decision boundary. Any data point that falls on either \( H1 \) or \( H2 \) is defined as a support vector (Han et al., 2012). The purpose of establishing the largest possible margin is to prevent classification errors on
previously unseen data. This is due to small margins being weak at generalizing on unseen data and tending to overfit (Tan, Steinbach, & Kumar, 2006). A visual representation of a binary SVM classifier can be seen in Figure 4.

Some datasets cannot be properly separated using a linear hyperplane. In order to accommodate such data, SVMs apply the *kernel trick*. The kernel trick consists of projecting the data into a higher-dimensional space based on a basis function. Popular basis functions include polynomials and Gaussian. Once the data has been projected to a new dimension, a linear decision boundary is established within this new dimension (Vanderplas, 2017). A visual presentation of this can be seen in Figure 5.

**Naïve Bayes Classifier**

The Naïve Bayes Classifier is a supervised machine learning model used for binary as well as multinomial classification tasks. Naïve Bayes is a probabilistic classifier based on Bayes’ Theorem.
Classification based on Bayes theorem has the goal of deriving the probability of a class label conditional on observations. Bayes Theorem is expressed using the equation: (Vanderplas, 2017)

\[
P(\text{Label} \mid \text{features}) = \frac{P(\text{features} \mid \text{Label}) \times P(\text{Label})}{P(\text{features})}
\]

*Equation 2 – Bayes Theorem*

Where \( P(\text{Label} \mid \text{features}) \) is the conditional probability of a label given its associated features. Class probabilities are learned from a training set. These probabilities are then subsequently used to classify new instances (Taheri & Mammadov, 2013). The Naïve Bayes Classifier is based on the assumption that input features, \( x_i \), are mutually independent of each other given the class, \( c \): (H. Zhang, 2004). This is expressed as:

\[
P(x \mid c) = p(x_1, x_2, x_3, ..., x_n \mid c) = \prod_{i=1}^{n} p(x_i \mid c)
\]

*Equation 3 – Naïve Bayes Independence Assumption*

For text classification, this implies that the words in a corpus are considered to be independent of one another. Despite the independence assumption being unlikely in real-world scenarios, the model has proven to be useful and efficient in many text classification tasks (H. Zhang, 2004). Particularly in information retrieval (Lewis, 1998), text classification (Larkey & Croft, 1996; Lewis & Gale, 1994; Peng & Schuurmans, 2003) as well as for spam filtering (Hovold, 2005; Schneider, 2003).

Several variations of the Naïve Bayes are available in Sci-Kit Learn. For this paper, the Multinomial version was implemented as it is well suited for text classifications tasks (C. Müller & Guido, 2016).

**Logistic Regression**

Logistic regression is supervised probabilistic machine learning model used for binary and multinomial classification tasks in addition to regression analysis.

Logistic regression is a foundational machine learning model that has enjoyed popularity in a wide variety of applications, including applications in NLP, specifically text classification (Genkin, Lewis,
Logistic regression is a discriminative classifier. This implies that the model focuses on learning distinguishing features amongst the classes present in the data (Jurafsky & Martin, 2018). Given a training set of features, denoted $x_i$, the logistic regression classifier learns a weight, $w_i$, and bias term, $b$ for each input. The weight for each input determines how important that input is to the final classification. Once the weight and bias terms have been determined, the input is multiplied by the weight. The bias term is subsequently added. The result is a number $z$ that expresses the evidence for that input belonging to a class (Jurafsky & Martin, 2018).

$$z = (\sum_{i=1}^{n} w_i x_i) + b$$

*Equation 4 – Logistic Regression Class Evidence equation*

Keeping in line with the probabilistic nature of the model, $z$ is passed through a logistic function to ensure that it is mapped to a space ranging from $[0,1]$. For binary classification tasks, the logistic function in question is usually the Sigmoid function: (Herlau, Schmidt, & Mørup, 2018)

$$S(x) = \frac{1}{1 + e^{-x}}$$

*Equation 5 – Sigmoid Function*

For multinomial classification tasks, the Softmax function can be used (Murphy, 2012). The predicted class for each instance can be expressed as (Jurafsky & Martin, 2018):

$$y = S(z) = \frac{1}{1 + e^{-z}}$$

*Equation 6 – Logistic Regression Transformation of Result into range value*

For binary tasks, the final class can only be one of two classes. In order to accommodate this limitation, a threshold is defined to specify the classification of each input. Assuming that the classes in question are binary and numeric with values 0 or 1, the decision boundary can be defined as:
\[ \hat{y} = \begin{cases} 1, & y > 0.50 \\ 0, & \text{else} \end{cases} \]

_Equation 7 – Logistic Regression Decision Boundaries_

BERT-Based Models

In recent years, the field of NLP has experienced significant progress due to adaptation deep learning models. Model architectures based on neural networks have achieved state-of-the-art results on a variety of language tasks and pushed the field into a new era (Otter, Medina, & Kalita, 2019). Some of the most impactful recent developments within the field of deep learning NLP models will be described in this section.

Pre-training and Fine-tuning

One of the major advantages of deep learning NLP models is their ability to easily and relatively inexpensively be fine-tuned to perform a variety of different tasks. Many NLP models based on deep learning architectures are pre-trained on large datasets. This pre-training process means the model has learned a set of weights that only need slight adjustments to be adapted to tasks and data that the model has not been specifically trained on. This enables the end-user to easily and quickly adapt a model to a desired task without needing access to large amounts of data or computing power (Käding, Rodner, Freytag, & Denzler, 2016; Z. Wang, Bi, Wang, & Liu, 2019). This process is also known as Transfer Learning and allows the user to use a neural network without having to train all the weights from scratch (Weiss, Khoshgoftaar, & Wang, 2016).

Word Embeddings

Before the deep learning models are introduced, it is necessary to explain how computers handle text data. Computers and algorithms are not capable of understanding text. Due to this, text must be transformed to a representation that computers are capable of using for computational tasks.

Traditional approaches for establishing these representations include the BoW model and one-hot encoding schemes which represent words using term frequencies and sparse vector representations of word occurrence relative to a vocabulary, respectively (Goyal, Pandey, & Jain, 2018). While these approaches enable algorithms to perform mathematical operations on text, both approaches suffer from matrix sparsity and high dimensionality issues. In addition, these representations are incapable
of capturing semantic and contextual information relating to the input text. This means that words with similar meanings or associations are considered to be independent of each other (Zhao & Mao, 2017).

These issues were addressed with the concept of word embeddings. A word embedding is a dense feature vector representation of a word (Jurafsky & Martin, 2018). Word embeddings enable computers to capture syntactic and contextual information relating to words in a corpus (T. Yang & Li, 2018). By representing a word as a vector of real numbers, it becomes possible to represent the word in a vector space. This allows for mathematical operations such as calculating degrees of similarity or dissimilarity between words using methods such as Cosine similarity in addition to preventing the curse of dimensionality caused by sparse matrices (Taweh, 2018). An example of word embeddings visualized in a vector space can be seen in Figure 6.

Research has shown that performance of NLP models has greatly improved by using word embeddings that store word-related contextual information in a dense vector as opposed to traditional one-hot encoding and bag-of-words approaches (Turian, Ratinov, & Bengio, 2010).

Many different models can be used to create word embeddings. One such popular method is the Word2Vec model developed by researchers at Google in 2013 (Mikolov, Chen, Corrado, & Dean, 2013). Using a neural network architecture on an unlabeled corpus, Word2Vec can create word embedding vectors using two separate predictive tasks: Continuous Bag of Words (CBOW) or the
Skip-Gram approach. With the CBOW task, the model attempts to predict the current word in a sentence using the surrounding words. In the skip-gram approach, the model attempts to learn the surrounding words from the current word. While these approaches are highly effective in establishing word embeddings, they are not perfect. Word2Vec establishes a fixed word representation for each unique word independent of context. This implies that the model is incapable of properly capturing meaning and context of sentences, especially when polysemous words, i.e. words that have different meanings depending on sentence structure and context, are present (Y. Sun, Rao, & Ding, 2017).

A major advantage of models based on the Transformer architecture compared to other word embedding models such as Word2Vec and GloVe is their ability to produce contextual word embeddings. BERT and GPT-2 are capable of taking word order into consideration when creating word embeddings. This allows for adaptive and dynamic embeddings that fit to the contents and context of the sentence in which the word occurs (Ethayarajh, 2019). Contextualized word embeddings have proven to significantly improve results on several NLP tasks (Ethayarajh, 2019).

Transformer

An important innovation that has driven many of the recent performance improvement of NLP models is the Transformer architecture first described in late 2017 (Vaswani et al., 2017). The Transformer is a model architecture built around neural networks and attention mechanisms that enable algorithms to understand text more precisely than traditional approaches such as recurrent neural networks (RNN) and long short-term memory neural networks (LSTMs) (Karita et al., 2020). While the original Transformer implementation was designed for translation tasks, the architecture has proven to be adaptable to many other tasks such as text classification (X. Yang, Yang, Bi, & Lin, 2019). GPT-2 and the BERT-based models described in this section all utilize and incorporate concepts and components popularized by the Transformer architecture.

Components

The Transformer model is made up of two main components: an encoder component and a decoder component. The encoder component contains numerous encoder layers. Each encoder is comprised of a self-attention layer in addition to a feed-forward neural network (FFNN) layer. The decoder component shares a similar structure. The main difference being that each decoder inside the component also hosts an Encoder-Decoder Attention layer between the self-attention and FFNN
layer. The number of encoders and decoders inside each component is not fixed. The example shown in Figure 7 contains 4 encoder and decoder layers.

Figure 7 – Example of the Transformer model used on a translation task

**Encoder**

The input sentence is fed to the first encoder layer in the stack. For each token in the input, a word embedding is created. A positional encoding vector is also created. This tells the model in which position each token belongs. The word embedding vector and positional vector are combined using element wise vector addition (Barnett, 2019). The combined vector is passed through the self-attention layer. This is done to determine which features in the input are most important for the word of interest. Each output of the self-attention layer is then independently passed to a FFNN for further processing. The output of this FFNN is passed on to the next encoder block in the stack where the same process of self-attention and FFNN processing is applied. Once the input has passed through all individual blocks within an encoder, the output is passed to the decoders for further processing (Vaswani et al., 2017).
Self-Attention

Self-attention is a mechanism that allows the Transformer model to better understand how words in a sentence are lexically and semantically related. It is best explained with an example:

*The cat did not go for a swim in the river because it was hungry*

*Example 1 – Sentence showcasing presence of anaphora*

A human can easily tell that the word “it” refers to the cat, not the river. Self-attention enables the model to map and remember these relationships between the words which results in improved performance and understanding of text (Vaswani et al., 2017). This enables Transformer models to achieve a high level of sentence information by using information from non-adjacent words. The self-attention for each word in the input is calculated using a series of basic linear algebra operations. For all word embeddings inputted to the model, a query (representation of a currently processed word), a key (so-called labels of the other words in a sequence) and a value (actual representation of other words in the sequence) matrices are created by multiplying the word embedding by three weighted matrices initiated during the training process (Alammar, 2018).

Following this, a score is calculated for each word in the input sequence. The score for each individual word is compared to the score of all other words in the input. This score is used as a metric to determine how much self-attention should be placed on that specific word in the input. In order to calculate the scores, the key matrix and query matrix are multiplied. Once done, the result is divided by square root of dimensions of the key vector. Finally, the resulting value is passed through a Softmax function to ensure that all results are in the range [0,1] and add up to 1. From this, the Softmax scores are multiplied by the value matrix, the output of the self-attention layer for the encoded sentence has then been calculated (Alammar, 2018). To ensure that too much attention is not applied to a single word, Transformers make use of multi-headed attention. This means that the scoring process is conducted several times with different matrices. The results are then concatenated, multiplied by a weight matrix and passed to the FFNN layer (Vaswani et al., 2017).
Decoder

The purpose of the decoder is to produce the final output of the model. For translation tasks, this is the input sentence translated into another language.

The decoder is built similarly to the encoder with the main difference being that it contains an Encoder-Decoder attention layer in addition to the self-attention and FFNN layers. The Encoder-Decoder attention layer is identical to the Self-attention layer, with the exception that the Key and Value matrices it uses are inherited from the last encoder in the encoder stack. The decoder uses the Key and Vector matrices to get a better idea of where to put focus on the input sentence. In addition, the self-attention layer is masked. This limits the decoder from being able to “see” the next words that it needs to predict (Vaswani et al., 2017).

When training a Transformer model, an input sequence and its corresponding translation are provided to the model. The target sentence is transformed into the word embeddings with added positional information just like inputs in the encoder. Since the model is fed the correct output prediction, the model is fed a target sentence that is shifted one position to the right. In addition, in the self-attention layer a masking procedure is conducted. This is to ensure that the decoder cannot “see” the future words that it needs to predict. The decoder produces the correct output by predicting the next word in the sequence. Once a word has been predicted, it is added to the input of the decoder. The decoder can then use this word and all previously predicted words to predict the next word in the output sentence. This process is repeated one word at a time until a special “end of sentence” token is reached indicating that the translation is done (Gandhi, 2019). Just like in the encoder, the inputs to the decoder are passed through a self-attention layer and FFNN in order to process features related to the input. Once the input has passed through the FFNN layer, the final vector is passed to a linear layer and then a Softmax function in order to calculate the probability of the next token occurring at the specified position (Vaswani et al., 2017).

Transformers have proven to perform better than traditional NLP models due to their ability to calculate self-attention. In addition, Transformers can utilize parallelization as each input token is processed separately and not sequentially. This enables Transformers to utilize GPUs and thereby speed up computation times substantially compared to RNN and LSTMs (Hahn, 2019).
**BERT**

**Introduction to BERT**

Bidirectional Encoder Representations from Transformers (BERT) is a language representation model developed and open-sourced by Google in October 2018. BERT is pre-trained in an unsupervised manner on an unlabeled dataset of 16 GBs of text containing 3.3 billion words collected from the English Wikipedia and the BookCorpus dataset (Devlin, Chang, Lee, & Toutanova, 2019) and can be relatively easily and inexpensively fine-tuned for a variety of NLP tasks such as text classification, question answering, language inference and more (Devlin et al., 2019). BERT received widespread acclaim and attention in the NLP community upon its release due to its, at the time, state-of-the-art performance on benchmark NLP tasks such as GLUE, SquAD, MultiNLI among others (Devlin et al., 2019; C. Sun, Qiu, Xu, & Huang, 2019). Following its release, BERT and other models based on its architecture, have been considered state-of-the-art models within the field of NLP (Jawahar, Sagot, & Seddah, 2019; Kovaleva, Romanov, Rogers, & Rumshisky, 2019). The success of BERT has further been cemented since Google adapted its search algorithm to incorporate language understanding elements from BERT to improve search results (Nayak, 2019).

This section will introduce the BERT model as well as other models and architectures that are necessary to establish appropriate understanding of BERT.

**Model Requirements**

BERT is pre-trained on a fixed vocabulary learned from the corpora that the model is trained on. While this allows the model to recognize a majority of the most common words used in English, BERT includes functionality that enables it to handle words or characters that are not stored in the vocabulary. Before BERT can process any input text, it transforms it into sub-units by performing tokenization. The tokenization process splits the input sentence into separate units referred to as “tokens” (Manning et al., 2008). BERT utilizes a tokenizer based on the WordPiece model (Devlin et al., 2019). From the corpora that BERT is trained on, this tokenizer establishes a vocabulary of 30,000 tokens from the data. In addition to words, the tokenizer also established tokens for all unique single characters as well as sub-words. This approach ensures that unknown words are more likely to be represented by a representative token instead of [OOV] or [UNK] tokens used by other models to indicate that a word is not present in the vocabulary or unknown, respectively (Munikar, Sushil, & Aakash, 2019). For each entry in the vocabulary, a vocabulary ID is also assigned. This ID is used to
link the tokens to their respective word embedding vector. In addition to splitting input words into sub-units, all input sentences are prepended with a special [CLS] token and appended with a [SEP] token. For multi-sentence inputs, BERT only prepends a single [CLS] token. Sub words are prepended with two “#” characters. An example of the pre-processing conducted by BERT can be seen in Table 7.

<table>
<thead>
<tr>
<th>Original: Was Sen. John McCain a ‘Hanoi Hilton Songbird’?</th>
</tr>
</thead>
<tbody>
<tr>
<td>['[CLS]', 'was', 'sen', '.', 'john', 'mccain', 'a', 'hanoi', 'hilton', 'song', '#bird', '?', '[SEP]']</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original: The CIA paid two psychologists $81 million “to develop and run their torture program.”</th>
</tr>
</thead>
<tbody>
<tr>
<td>['[CLS]', 'the', 'cia', 'paid', 'two', 'psychologists', '$', '81', 'million', 'to', 'develop', 'and', 'run', 'their', 'torture', 'program', '.', '[SEP]']</td>
</tr>
</tbody>
</table>

Table 7 - Example of Pre-Processing by BERT on two sentences from the natural claims dataset

Every single token in the BERT vocabulary is associated with a word embedding vector. Each word embedding is a dense feature vector with the number of features equaling the number of hidden units in the chosen model. When applying the BERT model to a dataset, the user must specify the maximum sentence length that the model should accept. Current versions of BERT allow for a maximum length of 512 tokens per input instance. Sentences that are shorter than the maximum length are appended with [PAD] tokens until the maximum length has been reached.

BERT Architecture

One of the reasons why BERT is considered a state-of-the-art model is due to its ability to handle text in a bidirectional manner more efficiently than previous models. In order to properly understand how BERT achieves a bidirectional understanding of text, it is necessary to dive into some of the previous deep learning approaches that have been used to create language models.

One classical approach to solving language understanding tasks is to use recurrent neural networks (RNN). This type of neural network is designed to handle sequential data. This is useful for language tasks such as word predictions as the current word in a sentence often is dependent on one or more previous words in a sentence. RNNs can store a “memory” of the previous inputs in a sequence (Otter et al., 2019). For each input in a sequence, the RNN computes a hidden state which contains information on the importance of the input. This hidden state is used to process the next input in the sequence. In this way, the RNN is capable of remembering the importance of previous inputs in the
sequence to determine the appropriate classification of the current input (Staudemeyer & Morris, 2019).

A major disadvantage of RNNs is their inability to effectively remember long-term dependencies in text. This is due to the hidden states effectively becoming diluted as the sequence is being processed due to early results being overwritten by later results. This makes them less useful for NLP tasks such as word predictions as these often depend on information from various parts of the sequence in order to properly classify a later input (Otter et al., 2019). A solution to the long-term dependency problem was addresses using Long Short-Term Memory networks (LSTM). These networks are essentially expanded RNNs that add additional parameters designed to remember or discard certain pieces of information that might be of relevance in the sequence (Otter et al., 2019). While this enables them to perform better on long-term dependencies, LSTM are by no means perfect. A major issue with RNNs and LSTMs is their inability to process input data in a parallel manner. Modern deep learning models utilize the power of GPUs to process many streams of data in parallel. Sequential neural networks such as RNNs and LSTMs are unable to do this, leading to slow computation times (Bouaziz, Morchid, Dufour, Linarès, & De Mori, 2017). In addition to being slow, both RNNs and LSTM are inherently one-directional models designed to understand language in a single direction. Due to this, models based on these architectures are unable to properly capture meaning and context and thereby understand human language. One solution to the problem of unidirectionality is found using Bidirectional LSTMs (Bi-LSTM). Bi-LSTM networks are in essence two separate LSTM networks combined. Each network is tasked with processing the inputs from a different direction. This creates a model that is capable of analyzing text from left-to-right in addition to right-to-left (Halderman & Chiarello, 2005). This allows them to understand context in sentences better than traditional LSTMs and RNNs. In early 2018, Bi-LSTMs received increased attention with the release of the ELMo model that substantially improved NLP model performance by combining Bi-LSTMs with deep contextual word embeddings that are able to capture more contextual and semantic information related to words compared to static approaches such as Word2Vec (Peters et al., 2018). Despite these improvements and Bi-LSTMs showing promising results, the Transformer architecture released in late 2017 proved to perform even better than previous models.
The BERT model is largely based on the Transformer architecture with one main distinction. BERT does not include a Decoder component. Instead, the number of encoding layers is increased. Currently, two different versions of the BERT model exist. The details of these are outlined in Table 8.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Encoder Layers</th>
<th>Hidden Units</th>
<th>Attention Heads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Base</td>
<td>12</td>
<td>768</td>
<td>12</td>
</tr>
<tr>
<td>BERT Large</td>
<td>24</td>
<td>1024</td>
<td>16</td>
</tr>
</tbody>
</table>

*Table 8 – Attributes of Base and Large versions of BERT*

Encoder layers refer to the number of individual encoders that are included in the model. Hidden Units denote the number of weights in a single layer. Attention heads represent how many times the multi-head self-attention is performed. Both versions of the BERT model are provided as cased and uncased versions. The uncased version converts all input text to lower case. The cased versions take into account different cases in the input text (Petrova, 2019). This paper will use the BERT Base Uncased version of BERT.

A major advancement of BERT is its ability to leverage the Transformer architecture while still being truly bidirectional i.e. being able to understand text from a left-to-right as well as a right-to-left context. Traditional language models have not been bidirectional. One reason for this is that if a model would have to predict the words in a sentence from both directions, it would be able to “see” the word before predicting it and thereby rendering the training useless. BERT overcomes this issue because it is trained using the Masked Language Modeling (MLM) task. For the MLM task, 15% of the total input tokens are selected. 80% of these are replaced with a [MASK] token. 10% are replaced with a randomly selected token from the vocabulary. The last 10% remain unchanged. The goal is for BERT to predict the true vocabulary id of the [MASK] token positions based on its context (Devlin et al., 2019). BERT is also trained on the Next Sentence Prediction (NSP) task which it uses to establish relationships between sentences which has shown to improve performance on down-stream tasks. For this task, BERT is presented with two sentences. In 50% of the instances, the second sentence is a correct continuation of the first while in the other 50% the next sentence is randomly chosen from the corpus. BERT is tasked with predicting which sentence fits the context. This task
improves the model’s ability to understand text across sentences which has been proven to be beneficial to certain NLP tasks (Devlin et al., 2019).

**BERT-inspired Models**

In addition to the original BERT model, this paper will compare results to other models utilizing the BERT architecture. This section will introduce these models.

**RoBERTa**

RoBERTa is a modified version of BERT introduced by AI researchers at Facebook in mid-2019. RoBERTa is described as a “robustly optimized BERT” because of its expanded training process compared to the original BERT model. RoBERTa distinguishes itself from BERT in several ways. First of all, RoBERTa removes the NSP task from the training process. Instead, it improves the MLM task by dynamically changing the token masking pattern applied to the training data. In addition, the model is trained on a text corpus that is approximately 10 times larger than BERT’s thus increasing its size and diversity compared to the original model. The length of the sequences in the training data has also been increased (Y. Liu et al., 2019). RoBERTa has proven to outperform BERT in some NLP tasks (Y. Liu et al., 2019). Just like BERT, RoBERTa is also provided as base and large versions. This paper will use the base version of the RoBERTa model.

**DistilBERT**

Released in late 2019 by AI company Hugging Face, DistilBERT is a smaller version of BERT that is designed to provide comparable performance while substantially reducing the size of the model. In DistilBERT, the Transformer architecture has been shrunk resulting in the total number of layers in the mode being decreased by a factor of two. The resulting model is 40% smaller than the original BERT implementation while being 60% faster on inference speed and retaining 97% of its ability to understand language (Sanh, Debut, Chaumond, & Wolf, 2019). DistilBERT contains 66 million parameters compared to 110 for the base version of BERT and is trained on the same data as the original BERT model. The model does however leverage best practices in the pre-training process based on findings by the RoBERTa team (Sanh et al., 2019). The DistilBERT model is established on the principle of knowledge distillation.
Distillation, originally called model compression technique, is based on an intuition that a reduced and simpler model could be created using a large and computationally expensive model that has already been pre-trained on a large dataset by training the smaller model to approximate the function learned by the large model from the data (Bucilă, Caruana, & Niculescu-Mizil, 2006). Building on their work of model compression and coining the term ‘distillation’, (Hinton, Vinyals, & Dean, 2015) recognized the opportunity to transfer knowledge from a large model to a performance-optimized smaller model. In their approach, the large model, also referred to as the teacher, provides a probability distribution across the learned classes as the output of its last layer (Hinton et al., 2015). These class probabilities are learned from the training data and indicate the model’s ability to generalize beyond the training data (e.g. a language model predicting the next word for the sentence “I think this is the beginning of a beautiful [predicted word]” assigns a high probability to tokens such as [day] and [life] and a long list of tokens with low but non-zero probabilities for word such as [future, story, world]) (Sanh et al., 2019). Thus, using slightly adapted class probabilities from the larger model as “soft targets” in training of the smaller model is the fundamental idea behind model compression, or knowledge distillation (Hinton et al., 2015).

DistilBERT shares identical pre-processing requirements as the BERT Base Uncased model. This paper will use the DistilBERT Base Uncased version of the model.
This section will introduce the GPT-2 language model that will be used to produce artificial claims.

Generative Pretrained Transformer 2 (GPT-2) is a pre-trained predictive language model trained on 40GB of text. It is a Generative model as it was trained to generate tokens based on previous tokens in a sequence, Pretrained because of it being pretrained on 40GB of text, Transformer means that it uses the Transformer architecture and 2 indicates that it is the second version of the GPT model (Arasanipalai, 2019). The model was developed by San Francisco based research organization OpenAI (Openai.com, 2020). An overview of the different versions of GPT-2 can be seen in Table 9.

<table>
<thead>
<tr>
<th>Name</th>
<th>Release Date</th>
<th>Parameters</th>
<th>Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>February 2019</td>
<td>124M</td>
<td>12</td>
</tr>
<tr>
<td>Medium</td>
<td>May 2019</td>
<td>355M</td>
<td>24</td>
</tr>
<tr>
<td>Large</td>
<td>August 2019</td>
<td>774M</td>
<td>26</td>
</tr>
<tr>
<td>XL</td>
<td>November 2019</td>
<td>1.5B</td>
<td>48</td>
</tr>
</tbody>
</table>

Parameters denote the total number of weights in the model. Layers represent the total number of decoder blocks utilized by the model.

The strategy of staged release was employed by OpenAI due to concerns regarding potential misuse of the larger models (Solaiman et al., 2019). The organization claimed that releasing the model in stages would provide them with enough time and data to perform threat analyses, hence, identify potential misuses. However, some individuals within the AI field state that OpenAI’s primary goal was to gain extra media coverage, attract sensation seekers, and create extra hype around the research group by highlighting potential threats and impacts that GPT-2 might have (Lowe, 2019).

The uniqueness of GPT-2 lies within its capability to produce artificial text of high quality due to its size. OpenAI claims that its GPT-2 model managed to achieve state of the art results on 7 out of 8 language modeling datasets in a zero-shot setting (Radford et al., 2019). OpenAI teamed up with Cornell University to conduct a survey where people assigned a credibility score to text produced by the model. Participants rated text from the biggest model with a score of 6.91 of 10 providing evidence that the model can produce artificial text that people perceive as rather credible (Openai.com, 2019).
GPT-2 Model Architecture

The GPT-2 architecture is built solely on Transformer decoder blocks and outputs merely one token at a time, meaning that each token that is produced is subsequently added to the input and used as an input in the next prediction (Alammar, 2019). Thus, the model processes all previous tokens to predict the next token of the sequence, hence it utilizes an autoregressive approach for text generation.

Input

GPT-2 can produce text that contextually relates to input text from a user. Inputs can vary from a single word to multiple sentences (Alammar, 2019). The input provides the model with a foundation for the artificial text that it will produce. Once the input is provided, the input embeddings need to be looked up in the model’s embedding matrix. Positional encoding of tokens is also required to clearly indicate the order of tokens in a sequence (Alammar, 2019). A vector that is a combination of the word embedding and positional vectors is then forwarded to the first decoder block (Alammar, 2019).

Decoder Blocks

As opposed to traditional Transformer models, GPT-2 only uses decoder blocks. The number of blocks varies depending on the specific model implementation. Each block consists of two layers; self-attention and a FFNN. Once a token has been passed through a block, the resulting vector is forwarded to the next block (Alammar, 2019). An example of a decoder block can be seen in Figure 8.
The masked self-attention layer is a crucial part of each block as it facilitates the understanding of context within a sequence. Each Decoder block of GPT-2 includes a masked self-attention layer which means that the model merely looks at the previous tokens to understand the context of the sequence. Hence, GPT-2 is a unidirectional model. It assigns a relevancy score to every previous token which outlines how important a particular word is in explaining the context of a certain word before passing it to the FFNN layer (Alammar, 2019).

The product of each block within the model is subsequently passed to the next block until the last block has been reached. The output vector of the final block is multiplied by the word embedding matrix which creates token probabilities (Alammar, 2019). GPT-2 can consider multiple tokens based on their probability scores. Once a token has been predicted, the model will keep iterating until the last token is reached (Alammar, 2019).

**Utilization**

General language models like GPT-2 can be utilized in many different ways. As a model that produces high quality text and reaches state-of-the-art performance on several tasks in a zero-shot setting, GPT-2 can perform tasks like question answering, summarization or machine translation. Thus, it can be used by many entities to support many language tasks by creating writing assistants, dialogue agents,
unsupervised translation between languages or speech recognition systems (Openai.com, 2019). However, as previously mentioned, OpenAI expressed concerns about misuse of their model for malicious purposes such as generating fake news or the automation of production of abusive, phishing or spam content.

Evaluation Metric

One of the central elements of deep learning and machine learning is the task of determining a model’s ability to correctly classify instances from a dataset. This section will describe the accuracy score evaluation metric used to measure the performance of the baseline and deep learning classification models used in this paper.

Accuracy Score

The accuracy score is a simple measurement of the number of correctly classified instances relative to the total test size. For binary classification tasks, it can be expressed by the following (Metz, 1978):

\[
Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ instances}
\]

Equation 8 - Accuracy score for binary classification tasks

The Accuracy score metric will be used for all models as all datasets used in this paper are balanced.

Model evaluation

An integral part of all machine learning and deep learning projects is to evaluate the performance of one’s model. This is done to ensure that a model does not overfit to the data thereby making it incapable of generalizing on unseen data (C. Müller & Guido, 2016). A commonly used approach for evaluating the performance of a model is to train it on a subset of the dataset called the training data. Once this training process is done, the model is then fed an unseen data set called the testing data. Based on the results from the testing data it can be evaluated whether or not the parameters of the model should be tuned in order to help the model generalize. The ratio between training and testing data differs depending on the project. Most applications utilize between 70 – 80% of the data for training and 30 – 20% for testing (Géron, 2019). For the tasks conducted in this paper, a split of 70% training and 30% testing was chosen. To ensure that the same exact subset of data would be chosen by the model on any consecutive runs, the random state
hyperparameter was defined. The random state hyperparameter was set to 42 for all models. This ensures that results can be validated and checked across consecutive runs.

Model Optimization
To ensure that the implemented models achieve the best possible results, it is important to find the optimal combination of hyperparameters for the input data. One such method of doing this is to utilize the Gridsearch module found in SciKit-Learn (Pedregosa et al., 2011). This method conducts a cross-validation on the training data to find the values for the individual hyperparameters that provide the highest score. These values are then subsequently used for the classification task.

Creation of Artificial Text Using GPT-2

Machine Produced Text
The GPT-2 model and its code have been made publicly available to use by researchers by OpenAI.7 OpenAI claims that the robustness and the worst-case behavior of the model are not fully understood, thus, the best practices for selecting of optimal hyperparameters are yet to be identified.

Text Production
GPT-2 requires input in order to determine the style of text that it is supposed to produce. In addition, GPT-2 can utilize several hyperparameters to tune the production process. The exact influence of the hyperparameters on the final output has yet to be thoroughly researched. Due to this, it was not possible to determine the most optimal hyperparameters for the task at hand. OpenAI suggests evaluating GPT-2 for each task individually and selecting optimal hyperparameters accordingly (Radford et al., 2019). However, this approach requires extensive testing and laborious evaluation which was out of the scope of this paper.

Therefore, the hyperparameters used in this paper were defined with an intention to produce claims similar to the claims found in the natural claims dataset. The claims from the natural claims dataset were used as input prompts for GPT-2. An overview of the chosen hyperparameters can be seen in Table 10.

7 https://github.com/openai/gpt-2
This paper employed the, at the time of writing, largest publicly available version of GTP-2. Utilization of their biggest model should ensure outputs of the highest quality. Furthermore, the seed was set to “none” in order for the model not to reproduce the results from previous runs. Additionally, the number of samples to produce at a time as well as batch size were set to “1”. Due to the possibility of producing incomplete sentences, the length of produced samples was not limited. The temperature hyperparameter controls degree of randomness with 0 being the lowest and 1 the highest. In order to avoid production of duplicates, the value of the temperature hyperparameter was set to “1”. The top_p hyperparameter represents a value of cumulative probability. Based on this probability, the most suitable tokens (the ones which cumulative probability exceeds the value of top_p) are considered for the prediction of the next word. Setting the value of top_p to a value higher than 0 will activate nucleus sampling which overrides the top_k hyperparameter specifying the number of possible tokens the model should consider.

Three examples of outputs produced by GPT-2 and their respective input prompts can be seen in Table 11.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Name</td>
<td>1558M</td>
</tr>
<tr>
<td>seed</td>
<td>None</td>
</tr>
<tr>
<td>nsamples</td>
<td>1</td>
</tr>
<tr>
<td>batch_size</td>
<td>1</td>
</tr>
<tr>
<td>length</td>
<td>None</td>
</tr>
<tr>
<td>temperature</td>
<td>1</td>
</tr>
<tr>
<td>top_p</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 10 - GPT-2 Hyperparameters

<table>
<thead>
<tr>
<th>Claim</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>I will tell you this, Russia, if you're listening, I hope you're able to find the 30,000 emails that are missing, Trump said.</td>
<td>We have four combat-ready brigades out of 40 in the U.S. Army.</td>
</tr>
<tr>
<td>While hepatitis A is rarely deadly, some have died from complications caused by it, including complications related to organ failure, hepatitis C and liver cancer.</td>
<td>A major Hepatitis A outbreak in San Diego has been pinned on undocumented immigrants there.</td>
</tr>
<tr>
<td>• Matthew Broderick, 36, the comedian and actress who will wed British actress Gwyneth Paltrow in a lavish ceremony later this spring, also told the newspaper that he is pregnant.</td>
<td>Brad Pitt, Jennifer Aniston Do Have “Baby Announcement,”</td>
</tr>
</tbody>
</table>

Table 11 - Examples of claims produced by GPT-2
Using the aforementioned hyperparameters, two distinct datasets were created. The first dataset was produced from claims that were all labeled as true in the natural claims dataset. The second was produced from claims labeled as fake. GPT-2 processed one claim at a time, produced a sample of unspecified length and moved on to another claim from the input. 16,241 claims were created from 1,243 randomly selected true claims and 15,236 claims from 1,089 randomly selected fake claims from the natural claims dataset. The intention was to produce enough claims to establish a substantial sample size used as additional training data as due to a lack of sufficient computing power, generating samples using every claim from the natural claims dataset as a prompt and, subsequently selecting the most appropriate ones was infeasible.

**Fact-Checking**

With the increasing focus and emergence of fake news in recent years, fact-checking has become relevant and desirable by organizations and individuals aware of the potential threats of fake news. As a result of demand, numerous fact-checking websites have emerged since the early 2000s (Aspray & Cortada, 2019).

Fact-checking constitutes a process of assigning a veracity label to claim in a particular context (Vlachos & Riedel, 2015). Thus, fact-checkers review a claim, its context as well as date and time in order to be able to successfully assess the veracity of the claim. Context and date often happen to be crucial features of a claim as some claims might only be referred to as true when complemented with additional information or in other cases the information that a claim was supported by in the past, is now outdated (Vlachos & Riedel, 2015). The aforementioned cases are instances when the veracity assessment might not result in an unambiguous verdict, hence, fact-checking is not always a binary classification (Vlachos & Riedel, 2015).

The output of a fact-checking process is a verdict. The verdict represents a label (true, false, mostly true etc.) that clearly specifies the claim’s veracity. Furthermore, each verdict should be backed up by supporting evidence which outlines the reasons for assigning a particular value as well as sources for the information used (Vlachos & Riedel, 2015). In this paper fact-checking plays a key role in determining the veracity of the GPT-2-generated claims, hence, enables assigning an accurate veracity label to the produced claims.
Fact-Checking Artificial Claims

Of the over 15,000 produced fake sentences, 3,000 were randomly chosen and their veracity checked. It was first checked whether or not the claim could be verified. If an entity, person name or quote was present in the sentence then it was deemed suitable for verification. All verifiable claims were googled to find supporting evidence. If the machine produced claim could not be supported by evidence available online, the claim was labeled as fake. Furthermore, some of the claims were disproved by information from trustworthy sources and only claims supported by evidence were labeled as true. The sources of information used for fact-checking varied from newspaper websites to Wikipedia articles. The complete output of the verification and fact-checking process can be found in Appendix 3 – Manually Verified and Labeled Outputs From GPT-2. Of the 3000 total claims, 690 were deemed suitable for verification. Of these, 649 were labelled as verifiably fake and 41 as verifiably true. Human-like grammar and writing style made many of the claims seem very similar to human produced text.

Following the verification process, the artificially produced true and fake claims were added to the natural claims dataset. The dimensions of the concatenated dataset can be found in Table 12.

<table>
<thead>
<tr>
<th>Label</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake</td>
<td>11278</td>
<td>51.4</td>
</tr>
<tr>
<td>True</td>
<td>10670</td>
<td>48.6</td>
</tr>
<tr>
<td>Total</td>
<td>21948</td>
<td>100%</td>
</tr>
</tbody>
</table>

*Table 12 - Dataset dimension with manually fact-checked artificial claims added*

This resulted in a dataset with 51.4% fake and 48.6% true claims. Despite the slight imbalance of class labels, the overall dataset was still considered balanced.
Results

This section will present the results from the analysis and the performance of the implemented models.

The baseline and BERT-based models were applied to three separate binary classification tasks:
1. The natural claims from the natural claims dataset
2. The natural claims from the natural claims dataset in addition to 10,000 randomly chosen artificial claims produced by GPT-2.
3. The natural claims from natural claims dataset in addition to 690 artificial claims produced by GPT-2 that had been manually verified and fact-checked.

In order to determine the hyperparameters that provided the highest scores for each individual model, an extensive grid search process was conducted. The process was conducted on the natural claims task only. This grid search process was conducted on the most relevant hyperparameters for each model using 10-fold cross validation on TF-IDF in addition to BoW (CV) representations of the input data. In addition, several combinations of N-grams on ranges of words and individual characters (char) were tested. The highest accuracy scores achieved from the grid search process can be seen in Table 13.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.6878</td>
<td>0.6887</td>
<td>0.6768</td>
<td>0.6901</td>
<td>0.6961</td>
<td>0.6976</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.6989</td>
<td>0.7013</td>
<td>0.6948</td>
<td>0.6887</td>
<td>0.7053</td>
<td>0.7042</td>
</tr>
<tr>
<td>Linear SVM</td>
<td>0.6893</td>
<td>0.7024</td>
<td>0.643</td>
<td>0.6932</td>
<td>0.6898</td>
<td>0.6984</td>
</tr>
<tr>
<td>Non-linear SVM</td>
<td>0.498</td>
<td>0.498</td>
<td>0.5083</td>
<td>0.498</td>
<td>0.498</td>
<td>0.498</td>
</tr>
<tr>
<td>Average Score</td>
<td>0.6435</td>
<td>0.6476</td>
<td>0.6307</td>
<td>0.6425</td>
<td>0.6473</td>
<td>0.6495</td>
</tr>
</tbody>
</table>

Table 13 - Results from baseline models

From the table, several observations can be made. First of all, it can be observed that the Linear SVM implementation achieves higher scores than the non-linear implementations that were tested. The non-linear SVM implementations were therefore not used for subsequent tasks. In addition to this, it can also be observed that the TFIDF representation generally outperforms the CV representation. One
exception to this is the Naïve Bayes model which tends to achieve the best average performance on the CV representation. Based on the average score across all six variations of word encodings, it was decided that for all subsequent tasks the TFIDF Word Gram [1,3] word representation would be used as it achieved the highest average score across all models. An overview of the specific hyperparameters used for each model can be found in Appendix 2 – Optimal GridSearch Hyperparameters for Baseline Models.

The BERT, RoBERTa and DistilBERT models were all implemented using the Simpletransformers Python library (Rajapakse, 2020). This library allows for quick and easy customization of a wide array of Transformer-based models pre-trained from Hugging Face. The aforementioned models were implemented with the fine-tuning hyperparameters shown in Table 14.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epochs</td>
<td>10</td>
</tr>
<tr>
<td>Max Length</td>
<td>512</td>
</tr>
<tr>
<td>Batch Size</td>
<td>15</td>
</tr>
<tr>
<td>Lowercase</td>
<td>True</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>5e-5</td>
</tr>
</tbody>
</table>

Table 14 – Fine-tuning arguments for BERT-based models

Epochs denote the total number of times a full passthrough is conducted. The batch size hyperparameter determines how many samples will be passed through the network at a time. When one batch has been processed, the model’s loss is evaluated. The model’s internal parameters are then updated accordingly for the next epoch.

Due to a lack of sufficient computing power for extensive testing of optimal combinations of training arguments for each individual model, the learning rate chosen for the BERT-based models is the learning rate that the creators of BERT outline in their paper as being optimal for most tasks (Devlin et al., 2019). Increasing the value of the batch size hyperparameter proved to significantly improve the results achieved by some of the models. The batch size was therefore set to the maximum possible value that could run on the hardware that was available for testing. All models were run in a Google Colab Pro notebook with a Tesla P100 GPU with 16GBs of VRAM.
Results on Natural Claims Dataset

The results from the first task are shown in Table 15.

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td>0.6984</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>0.7042</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.6976</td>
</tr>
<tr>
<td>BERT</td>
<td>0.7035</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.7094</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>0.6984</td>
</tr>
</tbody>
</table>

Table 15 - Scores for Task 1: Claims from natural claims dataset only

The RoBERTa models achieves the highest overall accuracy score. Interestingly, the Naïve Bayes implementation achieves results that are comparable to RoBERTa and better than BERT and DistilBERT. The larger RoBERTa model achieves the highest score while the smaller DistilBERT model achieves a slightly lower score compared to the original BERT model. Given that RoBERTa is trained on more data than the original BERT model and that DistilBERT was designed to achieve results similar to BERT while utilizing faster inference, these results tend to follow a pattern related to size of the model and training data.

Results on Natural Claims Dataset and Non-Fact-Checked Artificial Claims

In order to determine the effects adding artificially produced text data based on naturally occurring claims has on the classification accuracy of the models, a new subset of training data was added to the existing training set. For this process, a random subset of claims from the natural claims dataset were selected and passed to the GPT-2 model as input prompts. From the GPT-2 outputs, 5,000 fake and 5,000 real claims were randomly selected. These were added to the existing training data from the natural claims dataset.

The artificial claims were given the same label as the prompts they were produced from. To ensure that the models were trained on data that was not present in the testing set, the artificial claims that were added to the training set were all produced from prompts that only occurred in the training data. This was done to ensure that the model did not train on artificial claims that were produced from prompts in the testing data, as this could have given the model knowledge of the correct label. The
models were tested solely on claims from the balanced natural claims dataset. This was done to test how the performance of the models would be affected when additional training data was added. The results are shown in Table 16.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td>0.7015</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td><strong>0.7047</strong></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.7020</td>
</tr>
<tr>
<td>BERT</td>
<td>0.6991</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.6829</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>0.6953</td>
</tr>
</tbody>
</table>

*Table 16 - Model scores on natural claims dataset with 10,000 artificial claims added for training*

From these results it can be observed that the addition of 5,000 fake and 5,000 true artificial claims did not improve the performance of the BERT-based models. In fact, each BERT model performed worse compared to just being trained on the natural claims. On the other hand, the baseline models all performed better compared to being trained solely on the natural claims, all reaching accuracies of over 0.70 with Naïve Bayes achieving the highest score of 0.7047.

**Results on Natural Claims Dataset and Fact-Checked Artificial Claims**

As an alternative to naively producing 10,000 claims without checking their veracity, it was tested how including only verified and fact-checked artificial claims would affect the performance of the same models. A random subset of 3,000 artificial claims were verified and fact checked. Of these, 690 were given a veracity label. These were subsequently added to the training data in addition to the natural claims dataset. The testing data consisted solely of the claims from the natural claims dataset. The results from this are shown in Table 17.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td>0.7005</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td><strong>0.7058</strong></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.6980</td>
</tr>
<tr>
<td>BERT</td>
<td>0.6947</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>0.6967</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>0.7053</td>
</tr>
</tbody>
</table>

*Table 17 – Scores for models applied to natural claims concatenated with fact-checked artificial claims*
The table shows that Linear SVM, Naïve Bayes and DistilBERT performed the best with scores greater than 0.70. Compared to incorporating 10,000 automatically labeled artificial claims, the incorporation of the manually fact-checked claims slightly increased the average performance for some of the models. As with the previous task, the Naïve Bayes classifier performed the best. However, it still achieved a lower accuracy score than the best performing model RoBERTa (0.7094) in the first task when only natural claims from the natural claims dataset were used. In spite of its small size of only 690 instances, adding the additional manually fact-checked claims proved to increase the accuracy scores of all baseline models. Opposite to this, the accuracy scores of the two more robust models (BERT and RoBERTa) decreased, nonetheless, the smallest deep learning model used (DistilBERT) achieved a higher accuracy score than in the two previous tasks.
**Discussion**

The following section will discuss the results presented in the previous section in addition to highlighting specific limitations of the study, proposals for future work as well as reflections on the research conducted in this paper.

As recognized by Augenstein et al. (2019), pre-trained contextual encoding models are expected to provide classification models with more accurate embedding values for textual data compared to randomly initialized word embeddings, which in turn could be reflected in improved performance of the classifiers. The results of the naturals claims only task are consistent with this assumption, as the best performing model was RoBERTa. However, the traditional machine learning models stayed on par with the BERT-based models with Naïve Bayes outperforming the BERT-based models on several tasks. In the light of these results, two major considerations are presented in the following section.

The first major consideration is concerned with data size in relation to the models’ architectures. Neural networks, the underlying architecture of BERT-based models, notoriously require vast amounts of training data to outperform traditional machine learning models. Most machine learning models reach a point in the training phase where additional data does not lead to improved outcomes. Opposite to this, deep learning models benefit greatly from increased amounts of data (Alom et al., 2019). As Figure 9 from Alom et al. (2019) shows, traditional machine learning models can outperform deep neural networks when the size of training data is limited.

![Figure 9 - Data requirements relative to performance for deep learning and machine learning models](image-url)
This issue is in part addressed by pre-training the BERT-based models on enormous amounts of training data, nevertheless, it does not eliminate the need for domain-specific data to fine-tune the pre-trained models to a specific task. Therefore, it can be discussed if the 21,258 training examples contained in the natural claims dataset constituted a proper size to utilize the potential of the BERT-based models. As a comparison, the SQuAD v1.1 NLP benchmark, which BERT was originally evaluated on, uses a dataset of 100,000 question-answer pair examples for fine-tuning, i.e. almost five times as much data as used in this paper (Devlin et al., 2019). Furthermore, the creators of RoBERTa stated that BERT was significantly undertrained, which can provide an indication as to why RoBERTa outperformed BERT and scored the highest in the analysis (Y. Liu et al., 2019). Notably, Devlin et al. (2019) found that BERT Large outperformed BERT Base on all tested tasks, particularly when little training data was available for fine-tuning. However, due to limitations on the available computing power, this paper could not test the performance of the BERT Large model on the tasks presented in this paper.

Alternatively, using the original, label imbalanced version of the MultiFC dataset would have led to an increase in training data relative to using the label-reduced natural claims dataset. This would lead to major class imbalance which would have required the use of different evaluation metrics.

Another crucial consideration related to the architecture of the BERT-based models and their performance compared to the baseline models relates to text pre-processing requirements for the different models. The BERT-based models have vastly different pre-processing requirements compared to the baseline models presented. The BERT-based models utilize a built-in tokenizer that automatically transforms text to the required representations. Due to this, the end-user is not required to perform any pre-processing of the input text before passing it to the model for fine-tuning. Opposite to this, the baseline models require the user to manually decide on and evaluate different pre-processing steps to test which combinations provide the most satisfactory results. Therefore, the differences between the text pre-processing performed for baseline models and by BERT-based models’ built-in functionality might have affected the final performance scores, although due to time constraints an analysis of potential differences was not conducted as a part of this paper.

The second major consideration relates to the classification task itself. Although all the applied models performed significantly better than random guessing (50%) on the natural claims dataset
without incorporation of any supporting evidence, there is still room for improvement. The results indicate that the models are capable of capturing some linguistic features or patterns that they can utilize to successfully distinguish between fake and real news. However, an accuracy score of slightly above 70% for a robust model like RoBERTa signals that the inherent nature of the fake news detection task is complex and requires further research. This research should, in particular, focus on how to generalize performance on out-of-domain datasets, where large pre-trained language models, such as BERT-based models, should have an advantage over traditional machine learning models due to pre-training.

An important consideration relating to the application of the presented models for automatic fake news detection is the issue of inference time i.e. the time it takes for a model to assign a label to a provided claim. As mentioned in the Economics of Fake News section, search engines and social media platforms act as digital curators of news. These platforms have an incentive to implement automatic fake news detection algorithms in their services to protect their brand and reputation. However, as the inference speed of large deep learning models is slow with even DistilBERT reporting inference speeds of about 410 seconds for a sentiment analysis task (Sanh et al., 2019), the digital curators who rely on providing fast access to news for their users might be disincentivized to utilize such models as increasing the time a user has to wait for results to be returned in a search engine could compel the user to use an alternative provider, thus losing revenue for the search engine to its competitor. Therefore, it is not currently viable to implement automatic fake news detection systems based on the models presented in this paper.

While the results of the three tasks indicate that the models are capable of capturing some kinds of syntactic cues in text which enable them to correctly classify most of the instances, they also prove that the models are incapable of properly classifying a large share of the instances in the dataset. It can therefore be argued that a system based on the models, data and computing power presented in this paper should not be deployed as a method for automatic detection of fake news in a real-life scenario as it would result in too many wrongful predictions and thereby be too unreliable. Even though requirements for accuracy vary between models, datasets and specific tasks, it is hard to argue that a computationally expensive system which can correctly classify only around 70% of instances and performs on par with baseline models should replace real humans with expert knowledge for the process of fact-checking. Due to this, there is still extensive research and testing to conduct within
this field to determine if it is feasible to implement automatic systems that are capable of determining if a claim should be labeled as fake or true.

Regarding the 2nd classification task, the incorporation of the 10,000 artificial claims led to decreased scores of every BERT-based model but improved the accuracy scores of the baseline models. Some important aspects that need to be considered when discussing these results are the claims themselves and the methodology for their generation. A main assumption underlying the process of producing text was the notion that using claims with a particular label (fake or true) as prompts would produce claims of the same veracity. However, this paper showed that providing GPT-2 with fake claims as an input can result in the model producing claims that are verifiably true.

While the text produced by GPT-2 shows an overall impressive level of coherences, syntax, structure and grammar, many instances that were produced can clearly be identified as being produced artificially. These findings are in agreement with the conclusions of the creators of GPT-2 who found, when testing GPT-2 on a summarization task in a zero-shot setting, that despite the produced text being similar to summaries, GPT-2 confused details such as numbers or items, which led them to conclude that GPT-2’s performance is insufficient for use in real-world settings (Radford et al., 2019). An example of this can be seen in Example 2.

*Vanna White and Shark Tank Return to the Finishing Line. A "very dangerous" man has been arrested after four people were stabbed on a train in Melbourne's east.*

*Example 2 - Example of output produced by GPT-2*

Although the claim is seemingly coherent in its wording, the text combines two unrelated topics into a single claim. Based on this, it is obvious that the model has no inherent understanding of the text it produces. Even though this method can provide classification models with more training data, the approach to labeling the artificial claims in this manner should be regarded as inadequate. For this reason, an extensive qualitative analysis of the labels’ veracity is required.

In relation to the 3rd classification task, BERT and RoBERTa did not benefit from the addition of the 690 manually fact-checked GPT-2 produced claims to the training dataset. All baseline models in addition to DistilBERT showed improved results relative to the natural claims only task. Of the 3
classification tasks, Naïve Bayes proved to consistently achieve the highest scores of all the baseline models. A similar pattern was not evident from the results of the BERT-based models.

Given that deep learning models generally perform better when the number of training data instances increases, it is difficult to pinpoint exactly why the BERT-based models performed worse when fine-tuned on the artificial claims in addition to the natural claims. One potential reason for this could be related to the attention mechanisms that are fundamental to the Transformer architecture. Given that some text instances produced by GPT-2 differ in style of writing and grammar from that of a human, the attention mechanisms could potentially experience difficulty allocating the attention properly. Future research should therefore be conducted to compare the performance of attention mechanisms on human and machine produced text to investigate if artificial text could be the cause of decreased performance.

While some of the models presented in this paper achieved improved accuracy scores after being trained on additional text data that was artificially produced, the methods for creating this data and their usefulness should be questioned.

Having discussed the main findings, the answer to the proposed research question is provided.

*How do BERT-based language models compare to traditional machine learning models in NLP on the fake news detection task on human and machine-produced text?*

From the findings presented in this paper, it is clear that determining exactly how Transformer-based language models can be used for fake news detection and production is complex. While the results presented in this paper indicate that the models are capable of identifying some patterns in claims that enable them to properly classify a claim as fake or not, the same results also prove that the methods presented in this paper faced limitations. Based on the results of the analysis, RoBERTa outperformed the traditional machine learning models on the natural claims dataset task. Despite this, the results of the analysis showed that the addition of artificially produced training data led to improved performance of traditional machine learning models while resulting in lower scores for the BERT-based models. While the causes of this change in performance are difficult to identify, it is hypothesized that the lack of sufficient computing power, artificial claims that differed in style and
grammar from that of a human, and insufficiently large training data for fine-tuning BERT-based models are key contributors to the results presented in this paper.

Limitations

One major factor limiting the research conducted in this paper was the lack of sufficient computing power to utilize the full potential of the BERT-based models. While the Google Colab service does allow for easy and relatively cheap access to powerful GPUs, the resources available were still not sufficient for testing the full capabilities of the BERT-based models presented in the paper. This resulted in models that were not thoroughly optimized for the task at hand as they were tested to find hyperparameters that allowed them to achieve satisfying results without encountering memory issues. Initial testing showed that increasing the value of the batch size hyperparameter resulted in significant improvements to classification accuracy. If more powerful GPUs with more VRAM had been available, this paper would have been able to find the hyperparameters for each model that would result in the highest possible scores for each task.

While the natural claims dataset used in this paper is, at the time of writing, the biggest available dataset of naturally occurring fact-checked claims labelled with a veracity label (Augenstein et al., 2019), it would have been preferable to use a dataset that contained more instances and preferably covered more topics and entities and deriving from more sources. In addition, the dataset did contain some instances of low quality which is presumably a result of the initial scraping process.

One of the major drawbacks of using the GPT-2 model is its novelty and therefore lack of supporting information specifying the best practices and behavior for optimizing the model.

Furthermore, given the size of the model, running GPT-2 requires powerful computer hardware. Therefore, running the model multiple times using different hyperparameters and subsequently evaluating the output to better understand the behavior and influence of the hyperparameters on the output can be a lengthy process. Due to limited availability of computing power and time constraints, the chosen hyperparameters were believed to produce the most human-like output for the task at hand. Ideally, multiple tests and evaluations of different hyperparameter combinations would have been conducted to properly assess the optimal combination leading to the best results. In relation to this, it was not possible to produce artificial claims from all the claims included in the natural claims dataset.
In an ideal scenario, all the claims from the natural claims dataset would be inputted to GPT-2 as prompts. This would result in a much larger output that would then be analyzed individually by several people to determine their veracity and their usefulness in the classification task.

As aforementioned, GPT-2 and other language models have no inherent understanding of what is true or false. At their core, the models are based on probability distributions. The models simply output what they consider likely based on training data. In line with this, the claims that were produced using GPT-2 should preferably have undergone extensive fact-checking by experts with knowledge on the subject matter of the specific claim. While a simple Google search provides a decent indication of whether or not the mentioned entities did actually proclaim what is stated in the claim, there can be many intricacies and details in a claim which are necessary for a valid veracity rating that a simple Google search is not capable of providing.

**Future Work**

Given the limitations that have been presented in this paper, there are several opportunities for conducting future work within the field of fake news detection and production using language models. One seemingly obvious approach would be to employ the large version of the models presented in this paper as well as testing if the cased versions would lead to better scores. In addition, future work should experiment with using more powerful computer hardware that would allow for an extensive hyperparameter optimization process to investigate if better results could be achieved with the models presented in this paper. Using identical hyperparameter values for all models is rarely optimal and can likely lead to sub-optimal results that are misleading of the model’s true capabilities within a certain task.

One major addition that could potentially change the perspectives and results from this paper would be to utilize models that have been developed following the release of BERT. One such model is XLNet which was released in June 2019 (Z. Yang et al., 2020). This model is built on components of auto encoding as used by BERT and autoregression which has enabled it to outperform BERT on several benchmark NLP tasks. It is left to future work to determine if XLNet is useful for the task of detecting fake news.
The release of BERT has led to numerous spin-off models that attempt to optimize the BERT architecture in their own way. While two of these (DistilBERT and RoBERTa) were presented in this paper, many more exist and are currently being developed. It is left to future work to investigate the possibilities of utilizing more of these for an extensive comparative study.

While the results presented in this paper show signs that modern language models can potentially be used for fake news detection, it would be optimal to investigate how these models would perform on a fake news detection task that utilized a bigger label space. While the research presented in this paper is based on theoretical foundation limiting fake news to a binary classification problem, many news outlets and fact-checking websites utilize a more nuanced approach to labeling the veracity of claims. It would therefore be worthwhile to investigate how the models presented in this paper would perform on a multinomial classification task. This task would be dependent on a dataset of high quality and size that utilized a standardized rating-scheme for all claims, something that was not available at the time of writing this paper.

Future research could also investigate possibilities for incorporating meta-data and external textual features to Transformer-based models used for fake news detection.

Future research could also investigate the possibility of determining the degree of veracity for a claim. Instead of classifying a claim as either fake or not, the task could be to provide a numerical score that indicates to the reader the probability of the information being real or false, i.e. a claim with fakeness score of 90% would have much stronger indication of being plausibly unreliable compared to a claim with the fakeness score of 60%.

While the BERT-based models have shown to outperform existing models on a variety of language understanding tasks, the models inherently suffer from issues related to their size. As the size of the models grow, the inference speed decreases and the demand for more powerful hardware increases. While the models are capable of achieving promising results, they are usually slow and require more computing power than most researchers have access to. This limits the possibilities for future research and experiments with these models. Due to this, a production ready fake news detection system based on these models could potentially face several issues unless the models are optimized for faster inference and to run efficiently on smaller hardware setups. Several researchers have investigated
possibilities for reducing model sizes and improving inference speed while maintaining high results (Sajjad, Dalvi, Durrani, & Nakov, 2020; Xin, Tang, Lee, Yu, & Lin, 2020). Future work is needed to determine if these models could make an automatic fake news detection system feasible while maintaining performance.

Despite GPT-2 proving to produce some text that to a human makes little sense, the model is still capable of producing high quality, human-like text at a large scale. While this paper has focused on the applications of artificial text for improving training data in classifications tasks, future research should continue investigating the potential threats and risks associated with generating fake text and using this for real purposes (Zellers et al., 2019). This implies establishing strategies and methods for detecting artificially produced text in order to deter potential misuses of the technology by malicious actors.

Reflections

Training, updating, and operationalizing large pre-trained language models like BERT requires substantial computational resources, which is reflected in the financial cost of procuring access to hardware satisfying such demands as well as in the high energy consumption. To exemplify, Strubell et al. (2019) provide insightful analysis of the costs for deep learning models in NLP. Notably, they find that an improvement of 0.1 on the BLEU benchmark in English-to-German translation achieved by a NAS model, cost more than USD 150,000 in cloud compute resources without accounting for the carbon footprint of energy consumption (2019). In terms of carbon footprint of the state-of-the-art NLP models, training BERT can be compared to a trans-American flight (Strubell et al., 2019).

There is a less obvious consequence of the state-of-the-art NLP models increasing in size, with the latest model revealed boasting 17 billion parameters, which is more than ten times the size of the largest version of GPT-2 (Rosset, 2020). Increasing models sizes leads to increasing demand for computational resources that severely limits which researchers get to drive innovation as well as who gets to reap the benefits of it (Strubell et al., 2019). Given that the research on state-of-the-art language models is almost exclusively limited to well-funded teams of researchers working for large technological companies, it raises serious questions regarding the future of the research in the NLP field.
Throughout this paper multiple decisions influencing the approach of conducting the experiment were made. Given the novelty of the topic as well as models and methods employed, the best practices for utilizing the models have yet to be agreed upon and therefore, some of the decisions made had to be based on trial and error. Thus, despite striving for impartiality and employing critical realism that considers objects being independent of the human mind, certain level of bias within some of the procedures could not be avoided.

The label reduction process represents an example of introduced bias to the research, as this process was performed manually by the authors. More specifically, the process of label reduction consisted of acquiring information about a given non-binary label and subsequently assigning an appropriate binary value to a claim based on available information and the personal judgment of the responsible person. Therefore, modifying the dataset in this manner introduced bias that could not have been avoided, as this process was to a certain extent influenced by the authors’ perception of what is fake or true in alignment with the critical realist view that objects are understood by researchers based on their social conditioning.

Another instance of adding bias to the research occurred when producing text using the GPT-2 model. As aforementioned 15,236 claims from 1,089 fake and 16,241 claims from 1,243 true claims were produced using GPT-2. Due to a lack of sufficient computing power, producing text from all the claims from the natural claims dataset was not feasible.

However, the approach employed created additional bias as a result of not including all the natural claims dataset in the text production.

The process of searching for supporting evidence that was used to establish the veracity of a claim was performed by mimicking human fact-checkers. Initial identification of the verifiability of the claims was followed by a search for supporting evidence, making a judgement and a subsequent verdict. The process of fact-checking offered opportunity for additional bias. Firstly, proving the verifiability of the claims was dependent on the fact-checker’s notion of what could be considered verifiable. Secondly, finding appropriate supporting information from trusted sources might have been influenced by bias of the fact-checker since the appropriateness of information is a subjective concept. Thirdly, the verdict itself was based on the supporting information or lack thereof, and the
fact-checker’s judgement. The process of fact-checking was conducted based on the fact-checking best-practices aiming for objective assessment of the veracity of the claims, nevertheless due to the aforementioned reasons a certain level of bias was inevitable.

Apart from GPT-2, few models that are capable of producing human-like text exist. Grover, a model developed by the Allen Institute for AI, utilizes the same architecture as GPT-2, however, the models differ in output structure as well as input requirements (Zellers et al., 2019). For conditional text generation, GPT-2 requires only an input text that will be used as a basis for further text generation. Grover requires metadata such as domain source, date, authors or headline to produce an article (Zellers et al., 2019). The research group claims that text produced by Grover achieves a high level of trustworthiness and is perceived as more trustworthy than disinformation written by humans (Zellers et al., 2019). Nevertheless, this paper focused on researching the applications of a binary classification task on the text of a claim and a veracity label. Grover produces full articles with amounts of text that are out of the scope of the goal of this paper. In addition, a pitfall of Grover is its focus on propaganda articles which constitute just one type of fake news. Furthermore, Grover relates the text it produces to a context. Due to the requirement of external inputs, it was decided to leave an implementation of Grover to future work.

In addition to the claims and their associated meta data, the MultiFC dataset also contained a selection of snippet files associated with each claim. Each snippet file contained at most the 10 highest ranking Google search results for the claim at the time of dataset creation. The snippets therefore constitute external features that could potentially have improved the performance of the classification models. While research has shown that including external features can potentially enhance the ability of non-Transformer models to detect fake news (Augenstein et al., 2019; W. Y. Wang, 2017), it was decided not to incorporate the snippets for this project. This decision was made for several reasons. One of these reasons was the inherent limitation present in the BERT-based models of a token limit of 512. By incorporating the snippets into the claims, about 50% of the total instances in the dataset would exceed this limit. Apart from a decrease in the total number of data instances, this would have led to results that would be hard to directly compare as it would be difficult to determine exactly how much information was used from the snippets and claim of each individual instance. Another major limiting factor associated with using the snippets is the possibility of providing the models with the correct result. Given that the claims in the MultiFC have all been fact-checked by experts and that these fact-checked verdicts in addition to the claim in plain text are hosted on publicly available websites, it
seems unlikely that search snippets would not, for some instances, have included the webpage that fact-checked the claim itself. This could potentially have resulted in instances where the search snippet would reveal the true label of the claim, thereby allowing the model to “see” the correct answer to the classification task. This could lead to biased results and fake impression of the models’ accuracy and ability to effectively detect fake news. Due to these limitations, it was decided not to incorporate search snippets with the data used in this paper. It is left to future work to determine an optimal solution for incorporating external features such as snippets and to conduct experiments on their ability to improve classification scores.
Conclusion

This paper investigated the task of fake news detection using language models. The paper utilized the, at time of writing, biggest available dataset of naturally occurring expert fact-checked claims from a variety of sources. Following a label reduction process, the paper applied several different machine learning and BERT-based language models to a binary classification task on a balanced version of the dataset. The paper found that the models were capable of achieving accuracy scores above 0.70 with the RoBERTa model achieving the highest score of 0.7094. In addition, this paper utilized the GPT-2 model to produce artificial text based on the claims from the balanced dataset. The artificial text was added to the existing training data to examine its influence on the performance of the implemented classification models. The paper found that the addition of artificially produced text generally improved the performance of traditional machine learning models while leading to lower average scores for the BERT-based models.

This paper concludes that a real-life implementation of automatic fake news detection systems based on the models presented in this paper require further research due to existing limitations such as limited availability of sufficient quality training data, insufficient computing power, slow inference speed and sub-optimal configurations of the implemented Transformer-based models.
Bibliography


Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python*. O'ReillyMedia,
Inc.
Kleinberg, & B. Schölkopf, Eds.). Cambridge: Springer.
Boididou, C., Andreadou, K., Papadopoulos, S., Dang-Nguyen, D. T., Boato, G., Riegler, M., …
Proceedings*, 1739.
Psychology Review*. https://doi.org/10.1207/s15327957pspr1003_2
Bouaziz, M., Morchid, M., Dufour, R., Linarès, G., & De Mori, R. (2017). Parallel Long Short-
Term Memory for multi-stream classification. *2016 IEEE Workshop on Spoken Language
Brewer, P. R., Young, D. G., & Morreale, M. (2013). The impact of real news about “fake news”:
https://doi.org/10.1093/ijpor/edt015
sophistication improve language modeling? *Proceedings of the Thirty-Sixth Annual Meeting of
the Association for Computational Linguistics.*
ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*
https://doi.org/10.1145/1150402.1150464
Data Scientists* (First Edit; D. Schanafelt, Ed.).
https://doi.org/10.1017/CBO9781107415324.004
Canuma, P. (2019). The brief history of NLP. Retrieved from
https://medium.com/datadriveninvestor/the-brief-history-of-nlp-c90f331b6ad7
*Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and
Data Mining*. https://doi.org/10.1145/1401890.1401907
April 19, 2020, from World Economic Forum website:
https://www.weforum.org/agenda/2019/03/fake-news-what-it-is-and-how-to-spot-it/


https://doi.org/10.3389/fninf.2014.00014


Creating Neural Networks with Python. https://doi.org/10.1016/bs.host.2018.05.001
https://doi.org/SAIREC, South Africa International Renewable Energy Conference
Han, J., Kamber, M., & Pei, J. (2012). Data mining concepts and techniques (Third Edit).
https://doi.org/10.1109/ICMIRA.2013.45
James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning.
https://doi.org/10.1007/978-1-4614-7138-7_8
Jurafsky, D., & Martin, J. H. (2018). Speech and Language Processing - An Introduction to Natural


Nayak, P. (2019). Understanding searches better than ever before. Retrieved March 22, 2020, from blog.google.com website: Inputs to BERT%0D%0AComputers and algorithms are not capable of understanding or using human text. Due to this, text must be transformed to a representation that computers are capable of using for computational tasks. This task is referred to as feature e


Nordquist, R. (2020). What Is a Natural Language?


98
**Release strategies and the social impacts of language models.**


https://doi.org/10.1007/978-3-030-32381-3_17


https://doi.org/10.1145/860468.860471


Appendix

Appendix 1 – Label reduction of MultiFC Dataset

This appendix provides an overview of the full label reduction process of the MultiFC dataset. It is provided as an Excel sheet. The sheet contains the following:

- The original labels in the found in the MultiFC dataset.
- The reclassified label.
- The source website for each label.
- Count of websites using label.
- Link to website’s rating scheme (if available).
- Justification for re-labeling.

Appendix 1 can be found in the attachments folder with filename: “1 - MultiFC Label Reduction Process.xlsx”.

Appendix 2 – Optimal GridSearch Hyperparameters for Baseline Models

This appendix provides an overview of the highest scoring hyperparameters from the GridSearch process using the Baseline models.

The appendix is provided as an Excel sheet.

Appendix 2 can be found in the attachments folder with filename: “2 - Optimal GridSearch Hyperparameters.xlsx”.

Appendix 3 – Manually Verified and Labeled Outputs From GPT-2

This appendix provides an overview of the 690 claims produced using GTP-2 that were manually verified and fact-checked for a veracity label.

The appendix is provided as an Excel sheet. The sheet contains the following:

- The claim produced by GPT-2.
- A label denoting whether or not the claim could be verified.
- The veracity label given to the claim.
- Sources providing additional information backing up the label decision.
- Optional comment stating reasoning for label.
- The prompt from the MultiFC dataset that was used to produce the claim.

Appendix 3 can be found in the attachments folder with filename: “3 - Manually Verified and Fact-Checked GPT-2 Claims.xlsx”.