AUTOMATIC FAKE NEWS DETECTION ON TWITTER

TING SU

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy

SCHOOL OF COMPUTING SCIENCE

COLLEGE OF SCIENCE AND ENGINEERING
UNIVERSITY OF GLASGOW

APRIL 2022

Abstract

Nowadays, information is easily accessible online, from articles by reliable news agencies to reports from independent reporters, to extreme views published by unknown individuals. Moreover, social media platforms are becoming increasingly important in everyday life, where users can obtain the latest news and updates, share links to any information they want to spread, and post their own opinions. Such information may create difficulties for information consumers as they try to distinguish fake news from genuine news. Indeed, users may not be necessarily aware that the information they encounter is false and may not have the time and effort to fact-check all the claims and information they encounter online. With the amount of information created and shared daily, it is also not feasible for journalists to manually fact-check every published news article, sentence or tweet. Therefore, an automatic fact-checking system that identifies the check-worthy claims and tweets, and then fact-checks these identified check-worthy claims and tweets can help inform the public of fake news circulating online.

Existing fake news detection systems mostly rely on the machine learning models' computational power to automatically identify fake news. Some researchers have focused on extracting the semantic and contextual meaning from news articles, statements, and tweets. These methods aim to identify fake news by analysing the differences in writing style between fake news and factual news. On the other hand, some researchers investigated using social networks information to detect fake news accurately. These methods aim to distinguish fake news from factual news based on the spreading pattern of news, and the statistical information of the engaging users with the propagated news.

In this thesis, we propose a novel end-to-end fake news detection framework that leverages both the textual features and social network features, which can be extracted from news, tweets, and their engaging users. Specifically, our proposed end-to-end framework is able to process a Twitter feed, identify check-worthy tweets and sentences using textual features and embedded entity features, and fact-check the claims using previously unexplored information, such as existing fake news collections and user network embeddings. Our ultimate

aim is to rank tweets and claims based on their check-worthiness to focus the available computational power on fact-checking the tweets and claims that are important and potentially fake. In particular, we leverage existing fake news collections to identify recurring fake news, while we explore the Twitter users' engagement with the check-worthy news to identify fake news that are spreading on Twitter.

To identify fake news effectively, we first propose the fake news detection framework (**FNDF**), which consists of the check-worthiness identification phase and the fact-checking phase. These two phases are divided into three tasks: Phase 1 Task 1: check-worthiness identification task; Phase 2 Task 2: recurring fake news identification task; and Phase 2 Task 3: social network structure-assisted fake news detection task. We conduct experiments on two large publicly available datasets, namely the MM-COVID and the stance detection (SD) datasets. The experimental results show that our proposed framework, FNDF, can indeed identify fake news more effectively than the existing SOTA models, with 23.2% and 4.0% significant increases in F1 scores on the two tested datasets, respectively.

To identify the check-worthy tweets and claims effectively, we incorporate embedded entities with language representations to form a vector representation of a given text, to identify if the text is check-worthy or not. We conduct experiments using three publicly available datasets, namely, the CLEF 2019, 2020 CheckThat! Lab check-worthy sentence detection dataset, and the CLEF 2021 CheckThat! Lab check-worthy tweets detection dataset. The experimental results show that combining entity representations and language model representations enhance the language model's performance in identifying check-worthy tweets and sentences. Specifically, combining embedded entities with the language model results in as much as 177.6% increase in MAP on ranking check-worthy tweets,and a 92.9% increase in ranking check-worthy sentences. Moreover, we conduct an ablation study on the proposed end-to-end framework, FNDF, and show that including a model for identifying check-worthy tweets and claims in our end-to-end framework, can significantly increase the F1 score by as much as 14.7%, compared to not including this model in our framework.

To identify recurring fake news effectively, we propose an ensemble model of the BM25 scores and the BERT language model. Experiments conducted on two datasets, namely, the WSDM Cup 2019 Fake News Challenge dataset, and the MM-COVID dataset. Experimental results show that enriching the BERT language model with the BM25 scores can help the BERT model identify fake news significantly more accurately by 4.4%. Moreover, the ablation study on the end-to-end fake news detection framework, FNDF, shows that including the identification of recurring fake news model in our proposed framework results in significant increase in terms of F1 score by as much as 15.5%, compared to not including this task in our framework.

To leverage the user network structure in detecting fake news, we first obtain user embed-

dings from unsupervised user network embeddings based on their friendship or follower connections on Twitter. Next, we use the user embeddings of the users who engaged with the news to represent a check-worthy tweet/claim, thus predicting whether it is fake news. Our results show that using user network embeddings to represent check-worthy tweets/sentences significantly outperforms the SOTA model, which uses language models to represent the tweets/sentences and complex networks requiring handcrafted features, by 12.0% in terms of the F1 score. Furthermore, including the user network assisted fake news detection model in our end-to-end framework, FNDF, significantly increase the F1 score by as much as 29.3%.

Overall, this thesis shows that an end-to-end fake news detection framework, FNDF, that identifies check-worthy tweets and claims, then fact-checks the check-worthy tweets and claims, by identifying recurring fake news and leveraging the social network users' connections, can effectively identify fake news online.

Acknowledgements

I came to Glasgow more than 5 years ago. It was dark, gloomy, and rainy. It still is.I guess god felt guilty of putting everyone through such bad weather, they decided to make the scenery incredibly lively, and more importantly, filled the city with warm, kind, and friendly souls.

I met my supervisor, Iadh Ounis, and Craig Macdonald, through their famous course *Information Retrieval*. I can still remember the day I approached Craig, after one of the course; and the day I approached Iadh, after one of the seminar on Monday. At the time I didn't know what I was getting myself into, but it doesn't matter now, because of how supportive they are, academically and life-wise.

It was also here when I met my new group of friends, who are ambitious and humble, who have lent their helping hands countless times. They are: Anjie Fang, Xiao Yang, Xi Wang, Graham McDonald, David Maxwell, Zaiqiao Meng, Jarana Manotumruska, Amir Jadidinejad, Xiao Wang, Stuart Mackie, Richard McCreadie, Javier Sanz-Cruzado Puig, Thomas Jaenich, Hitarth Nrvala, and Sean MacAvaney. Thank you all for the support you gave me along my journey.

I also got to know some brilliant friends outside of the team, I hold them very dearly in my heart. Some of them lent me a shoulder to cry on, some of them gave me hugs when I was lonely. They are: Ole Stubben (and his family); Lorenzo Cinque; Thea Lin; Mircea Iordache; and Uma Zalakain (and her friends).

My mentors in Signal.AI showed me the career in industry. They are: Miguel Martinez; Dyaa Albakour; Jiyin He; and Raymond Ng.

I thank everyone mentioned here, and those that I encountered but did not mention here, for accompanying me through the past 4 years. Without you I won't be able to stand here and graduate.

I would also like to thank my reviewers – Dr Richard McCreadie and Dr Vinay Setty – for their time, and for providing me with critical discussions and suggestions.

Figure 1: My cat Leo, who cuddled me when I was tired or sad, and never stops loving me.

Additionally, Chinese Scholarship Counsel sponsored my study for 3 years. It is their financial support that enabled me to finish my study.

Finally, I want to say thank you to my whole family for supporting me for the past 27 year.

Dedication. (Is what you need.)

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

The Merriam-Webster English Dictionary [44] defines *information* as "*knowledge that you get about someone or something*". It is the medium of knowing and understanding the world. Without credible information, one cannot understand the world and society meaningfully and critically. There are various forms in which information is presented and spread. For example, books, archives, published papers, and news are all well used to inform the general public. Nonetheless, among all the forms of information, the news serves as the medium where information about recently occurring events are spread around and obtained by the general public. A piece of news can include highlights published in newspapers and tabloids, can be events that concern the local regions read out in radio broadcasts, can be fast-developing stories that are live on television, and can be anything an individual might want to know from online articles.

With factual information, there exist *non-factual* information, where one may genuinely misunderstand a concept, misremember an event, thus record and spread non-factual information as factual information. However, some may aim to deceive and mislead the information receivers and create false beliefs among the public, which we refer to as **fake news**. The deliberate act of circulating and spreading non-factual information can be malicious, and pose dangers to the society. The practice of creating non-factual information has been around for as long as recorded history. For example, evidence has suggested that creating misinformation and even deliberate disinformation have been a common practice since ancient Greece and the Roman empire [58], and the influence of such misinformation can be catastrophic and deadly, even to be the greatest empire in history. More recently, non-factual information has been used to discredit a range of political ideas, such as climate change [93] or vaccine safety and efficacy [42]. Some political scientists and psychologists [51, 65, 86] have

suggested that a paranoia personality and conspiracy mentality contribute significantly to creating and spreading fake news online.

Historically, the news has been mainly broadcast through traditional news media, such as newspapers, radio stations, and TV programs, which helps the general public effectively access information reported and selected by journalists. The credibility of these news sources is guaranteed by the journalists' names and the news organisation's reputation. However, with the rise of the internet, new ways of creating information have become widespread. Personal blogs, online forums, question and answer communities, social media platforms (e.g., Twitter, Facebook), etc., have become increasingly popular platforms to create information and present personal opinions that people from all over the world have access to it. The ease of producing information has contributed to the increasing popularity of online information created in the past decade and made the internet increasingly important for information consumers. There are roughly 500 million tweets created per day[1]; according to W3techs[2], WordPress has 65.1% of the content management system market share as of 2021, and on average sees 70 million new posts and 77 million new comments each month[3]. Moreover, Rosner et al. [149] suggested being able to remain anonymous online encouraged some individuals to produce and spread non-factual information, because online anonymous users are less likely to be identified and prosecuted.

The massive amount of information published online is challenging and time-consuming to verify. Thus, apart from reliable and credible news agencies, users may be exposed to non-factual information created by known or unknown individuals, independent content creators, agents using unverifiable identities, etc., which can be hard to identify if it contains misleading or malicious information [98].

The process of verifying if some information is factual or not is commonly known as *fact-checking*. Several websites are dedicated to fact-checking newly published information online. For example, Snopes[4] is a well-known website that covers a wide range of topics, where journalists manually examine the statements in articles of interest; Politifact[5] is a website mainly focusing on political topics and aims to inform the general public about political facts. These websites all enlist journalists to manually fact-check information submitted to them by their readers, and thus leave out information that is not noticed by the active users. However, despite the best effort of journalists and fact-checking websites, it is also not feasible for journalists to manually fact-check every news article, sentence or tweet online, given the amount of information generated daily.

[1] https://www.internetlivestats.com/twitter-statistics/
[2] https://w3techs.com/technologies/details/cm-wordpress
[3] https://wordpress.com/activity/
[4] https://snopes.com
[5] https://politifact.com

There are various formats where information spread, such as in news articles, personal opinions, podcasts, or speeches/debates. Yet most non-factual news concerning the public generally refers to statements regarding entities. For example, a person asserts that an event has happened; an article describes and summarises a governmental bill; a statement mentions an individual is ill, etc. We define such statements as *claim*s, which is also defined as "*an assertion open to challenge*" in the Merriam-Webster English Dictionary. In other words, a claim is a narrower version of information, where a news article discusses information regarding an event, including several claims related to the event. We argue that a claim is a smaller granulation of news, where complex information may be present. Thus, in this thesis, we only focus on detecting *non-factual claims*, where the small unit of information gives us a more precise objective in identifying it as factual or not.

Furthermore, *Fast & Furious Fact Check Challenge* [6] have identified four types of claims that need to be fact-checked: numerical claims are claims containing numbered facts; verification of quotes contains claims made about who said what; position statements are claims about personal stance; objects, properties and events are claims related to objective truth about existing events and entities. Each type of claim has a different focus and thus may need different types of models. For example, numerical claims require numerical evidence that may require calculation; verifications of quotes require finding original audio/video clips and textual evidence; positional statements are objective and can be interpreted differently. Thus, this thesis focuses on identifying non-factual information among object, property and event claims.

Specifically, let us consider a political debate such as the third 2016 US presidential debate between Hillary Clinton and Donald Trump, where Clinton has said: "... that you (Donald Trump) encouraged espionage (from Russia) against our people (the United States 2016 presidential election)". Such a severe accusation from Clinton toward Trump is important to fact-check, which is presented as if it is a fact that Donald Trump has colluded with Russia in meddling with the US election. Similarly, let us consider a Tweet from Trump that reads: "I WON THIS ELECTION, BY A LOT "after the 2020 presidential election had concluded that Joe Biden won the election with both more individual votes and electoral votes. In this case, such a publicised tweet claims that Donald Trump has won the election is contrary to the fact[7]. Therefore, this thesis considers the sentence in the debate transcripts and the tweet mentioned above as important claims that require fact-checking.

The internet era also changed how people access and spread information, as the internet enables people to access any information faster and easier than ever before and helps individuals find their preferred social groups and information [14]. For example, social media

[6] `https://www.herox.com/factcheck/teams`

[7] `https://www.fec.gov/resources/cms-content/documents/2020presgeresults.pdf`

platforms not only help users create information and post their own opinions, but also help users to spread and broadcast information and opinions to a broad audience. Moreover, users that want to obtain the latest news and updates on social platforms are thus easily presented with the information that their friends, families, and acquaintances spread, which increases their acceptance of such information[5, 176]. Consequently, according to Nielson et al. [130], up to 79% of the 18- to 24-year-old from 6 surveyed countries (Argentina, Germany, South Korea, Spain, the UK, and the US) now consider social media platforms alone as their primary news sources, as of 2020.

Specifically, Twitter has 211 million daily active users[8], and most world leaders and foreign ministries have an official Twitter account[9]. The prominence of Twitter allows individuals share information, news, and opinions with other people. Twitter is also widely studied among researchers, for the ease of obtaining data [43, 47, 48, 49, 64, 166].

Thus, this thesis concerns leveraging the large amount of information circulating on Twitter to identify fake news that is spreading online. Specifically, we aim to *build an **automatic fact-checking system** that is able to **identify** the most check-worthy **claims** within **tweets**, and to **fact-check** the identified check-worthy claims*.

In the following, Section 1.2 presents the motivations of our thesis; Section 1.3 presents the thesis statement; Section 1.4 lists the contributions of this thesis; the origins of materials are presented in Section 1.5; and finally Section 1.6 outlines the structure of this thesis.

1.2 Motivations

The current models and websites that aim to fact-check fake news mainly focus on fact-checking an entire news article which can contain a mixture of factual and non-factual information. This thesis argues that identifying non-factual articles is ambiguous, as more than one statement can appear in the same article, and not all statements are necessarily non-factual. Thus, we are motivated to reduce ambiguity by fact-checking claims, so that when fact-checking news articles, tweets, debate transcripts, we can inform the general public not only if they are misleading and/or non-factual, but especially which specific statement is misleading or malicious.

Moreover, with the notable impact of non-factual information being created, shared, and circulating online [148, 185], journalists and scientists are exploring ways to counter their impact. As mentioned before, `snopes.com` and `politifact.com` both employ journalists to manually fact-check each potential rumour, provide evidence for them and create

[8] `https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/`

[9] `https://www.statista.com/statistics/281375/heads-of-state-with-the-most-twitter-followers/`

articles that elaborate on the rumour. Currently, the manual fact-checking process requires general users to submit suspicious news articles/tweets. Then fact-checking journalists will determine if each article/tweet may have a high negative impact if it is non-factual and needs fact-checking. If it is confirmed that it needs fact-checking, the assigned journalists will look for evidence of whether the information provided by the article is fabricated, misleading, or true. Finally, fact-checked articles/tweets will be published on the fact-checking website, and the debunking articles may be posted on Twitter. However, this process of manual fact-checking is laborious. It not only requires a significant amount of journalists dedicating a vast amount of time to identify potentially non-factual news/articles/tweets and further fact-check them, but also requires the general public to be mindful of the information they encounter and be able to report potential non-factual information to the journalists. In order to reduce the labour requirement in the fact-checking process, this thesis argues that automatically removing tweets and claims from the fact-checking process, based on whether a tweet may contain non-factual information, can be effective in the fact-checking process.

Furthermore, Shin et al. [159] and Rosnow [150] have shown that rumours and fake news often resurface after they are identified as fake news. Some fact-checking websites such as `snopes.com` and `politifact.com` have published their historical data, where existing debunked fake news, fact-checked tweets, and articles are labelled with reasons for them being fake. These datasets are not only beneficial to researchers for studying the characteristics of existing fake news, but can also be used as reference datasets as to whether some fake news has resurfaced. Thus, we argue that it is important to incorporate external fake news datasets, in order to effectively identify reappearing fake news on Twitter.

Finally, some newly emerged fake claims primarily circulate on social media. The complex information within social networks, which includes network information regarding users and posts and the content information of posts, can be informative [83]. For example, researchers and journalists can look into the source of the claims and the social group that started the non-factual information, with the intuition that the *echo chamber* effect may amplify the collective bias and the belief in fake news. Thus, this thesis argues that the network structure of Twitter can be informative in identifying echo chamber effects, and thus can help determine if a check-worthy statement in a tweet is factual or not.

Overall, the goal of this thesis is to build a framework where individuals can search for a topic and receive trustworthy labels of the search results associated with the topic and query they searched for.

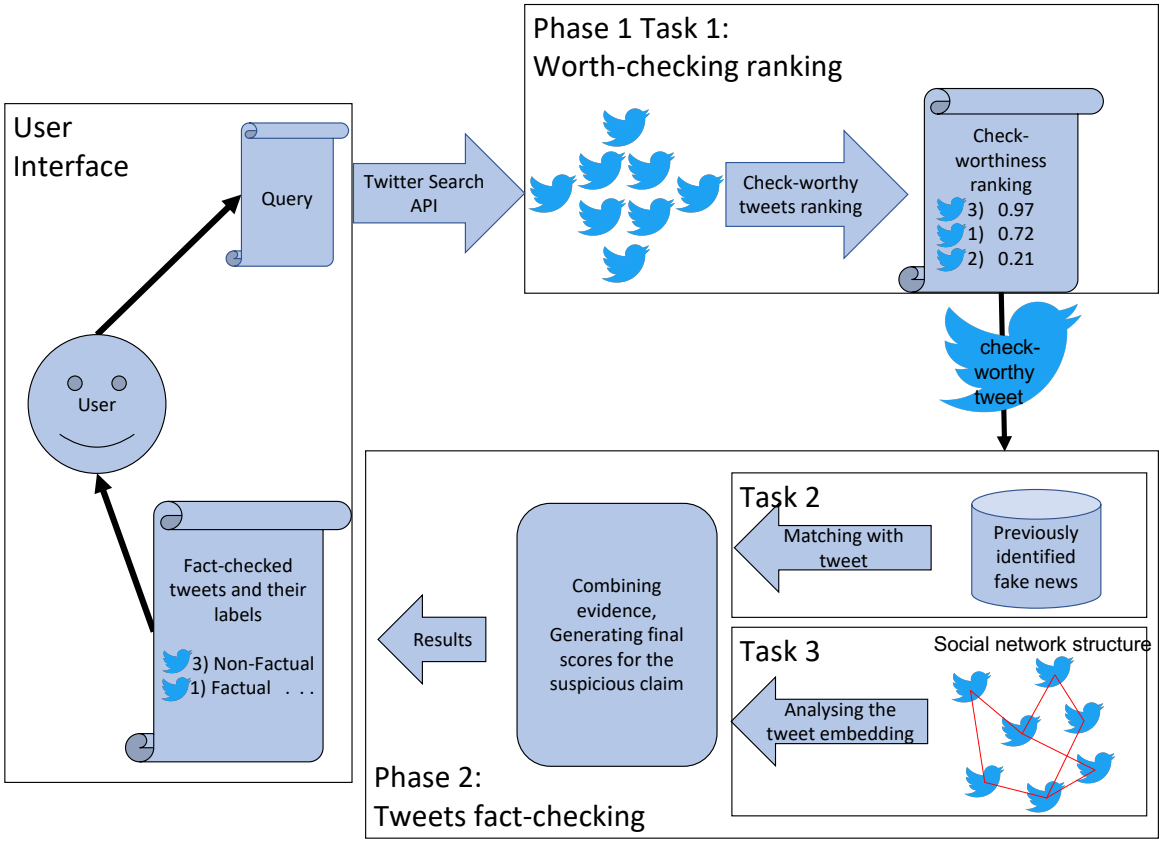


Figure 1.1: An overview of the proposed framework, from a user's perspective.

1.3 Thesis Statement

This thesis states that a two-phased Fake News Detection Framework (FNDF) (as shown in Figure 1.1) can achieve state-of-the-art performance in effective non-factual information identification on Twitter. The first phase, the worth-checking ranking (WCR) phase, consists of **identifying and ranking tweets and sentences** that contain **worth-checking claims**, where tweets and sentences are ranked based on their content's check-worthiness (Task 1). The second phase, the fact-checking (FC) phase, determines whether these check-worthy claims within tweets and sentences are factual or not, using among others, the information from **existing fake news datasets** (Task 2) and the **network** and **textual** information from the Twitter platform (Task 3). After these two phases, our framework will return the worth-checking claims within tweets and sentences, labelled as factual or non-factual. Specifically, in Task 1, we hypothesise that by analysing **embedded entities** in texts, we can more accurately **identify check-worthy claims**, from tweet content, articles, and debate quotes. Secondly, in Task 2, by comparing the targeted claim with existing non-factual news collection, we hypothesise that an **ensemble** textual model of a **BM25** model and a **deep neural network language model** can accurately classify if a targeted check-worthy claim is highly similar to any existing fake news, and thus is a resurfaced fake claim. Thirdly, in Task 3,

we hypothesise that **user network embeddings** trained with **unlabelled** user network data, can identify the echo chamber effects among users, and can be effective in identifying fake claims on **Twitter**. Finally, we hypothesise that by combining all components, our framework can achieve state-of-the-art performance in identifying fake news circulating on Twitter in an **end-to-end** fashion.

1.4 Thesis Contributions

The contributions of this thesis are four-fold:

1. In Chapter 4, we demonstrate that a model which combines textual representation with embedded entities can identify the check-worthiness of tweets more effectively than the current state-of-the-art text-based models. Specifically:

    - We propose a simple yet powerful model to represent sentences or tweets with rich entity information, by concatenating together a text model representation with entity pair representations;
    - Using the CLEF 2019, 2020 and 2021 CheckThat! Lab datasets, we show that our entity-assisted neural language models significantly outperform the existing state-of-the-art approaches in the classification task, and outperform the participating groups on the CLEF CheckThat! leader board in the ranking task;
    - We show that representing entity pairs with embeddings is significantly more effective than an existing recent technique from the literature that leverages the similarities and relatedness of the entities;
    - We show that simple deep neural language models cannot effectively identify check-worthy sentences and tweets;
    - Finally, our findings show that, among the various knowledge graph embedding models, ComplEx [180] leads to the best results. For instance, achieving results as good as the best performing system submitted to the CLEF 2019 CheckThat! Lab, without the need for labour-intensive feature engineering.

2. In Chapter 5, we show that existing fake news datasets can benefit fake news identification. Specifically:

    - We draw best practices in using deep learning language model representations to identify the relationship's between news titles, by comparing simple-embedding representations with BiLSTM and BERT;

- We examine how the traditional BM25 retrieval score can improve the performance of state-of-the-art deep neural network (NN) models;

3. In Chapter 6, we demonstrate that network information on social media, even in an unlabelled manner, is informative and can help our model identify newly emerged fake news. Specifically:

   - We propose a User Network Embedding Structure (UNES) model, which performs fake news classification on Twitter through the use of graph embeddings to represent Twitter users' social network structure. Compared to the existing approach of using user networks with handcrafted features, UNES does not require any pre-annotated data (e.g., user type (individual users or publishers), users' stance, and if they have engaged with fake news before);
   - We observe that the user embeddings generated by UNES exhibit a clustering effect between users who engage with fake news and users who solely engage with factual news, despite not having knowledge of whether the users have engaged with fake news before;
   - We also show that using the social network's user connections alone to build network embeddings, and using only users that engaged with the news when representing such news, can significantly outperform the existing state-of-the-art fake news detection approaches that use both textual features and complex social network features.

4. Finally, in Chapter 7, we show that our systematic framework, which combines multiple steps, can effectively identify tweets that state non-factual information. Specifically:

   - We demonstrate that our proposed FNDF framework in Figure 1.1. We show that the proposed FNDF is able to effectively detect fake news among tweets, using the 2 phase framework;
   - We show that our FNDF framework is able to detect fake news from another dataset, demonstrating the robustness of our framework;
   - We also show that every component of our framework has distinct functions in detecting fake news online.

1.5 Origins of Material

Most of the material presented in this thesis has appeared in several conference papers, or has been submitted to several journals, throughout the author's PhD programme. We list the

publications in chronological order, and we group our publications into several focuses:

- research concerning the use of existing articles and fake news datasets was published in SIGIR2019 [167] ;
- research concerning the ranking of claims in order to prioritise the most damaging claims was first published in CLEF 2019 [168], and then in another paper submitted to IPM [169];
- research concerning the impact of social media on fake news detection [171] has been submitted to the journal Online Social Networks and Media;
- empirical studies that quantify our framework's effectiveness [170] are to be submitted to ECIR 2023.

1.6 Thesis Outline

This thesis is structured as follows:

1. Chapter 2 presents the background for fake news detection, a comprehensive and extensive collection of recent studies on fake news detection and classification, as well as research advances used in this thesis. We also highlight the gaps within the existing literature that motivates our thesis.
2. Chapter 3 presents our proposed framework, the motivation for each component, and the tasks we aim to tackle in the framework. This chapter also formally defines the three tasks in the framework with designated terminology and equations.
3. Chapter 4 presents our research and experiments related to identifying check-worthy tweets and debates, where we show that entities are essential aspects that aid in detecting check-worthy tweets and debates more accurately than using only textual features.
4. Chapter 5 presents the research and experiments related to using external datasets for identifying resurfacing fake news. We show that we can identify recurring fake news by comparing targeted claims with the previously identified fake news headlines and claims. We demonstrate that ensemble models of deep NN language models and a traditional BM25 algorithm are more effective in identifying resurfacing non-factual claims than using deep NN language models alone.
5. Chapter 6 presents our proposed model that utilises unlabelled network information on Twitter for fake news identification. We show that user network embedding is an

important component in identifying the echo chambers that may facilitate the circulation and spreading of fake news online, which is more effective than textual features extracted from tweets and replies.

6. Chapter 7 presents the end-to-end evaluation of our framework, where we examine the effectiveness of our framework in identifying fake news on Twitter, conduct an ablation study, investigate the generalisation ability of our framework, and compare our framework to other fake news detection systems. We show that our proposed framework can more effectively identify tweets containing non-factual information than the individual components, and can be generalised to the unseen topics, in online fake news identification.

7. Chapter 8 concludes this work and highlights directions for future work.

Chapter 2

Background and Related Work

Fake news has existed throughout our written history, as history tends to repeat itself through time. In Section 2.1, we first introduce a brief background of fake news during the internet era and with the rise of social media platforms. In recent years, there has been an extensive body of research on fake news detection in social media. Thus, Section 2.2 discusses a wide range of research that aimed to tackle fake news identification. In Section 1.3, we argued that a two-phased and three-tasked end-to-end fake news detection framework can effectively identify fake news online. Thus Section 2.4 surveys the research w.r.t each task in our proposed framework. Then, we survey the commonly used methods for analysing the textual features of a news article (presented in Section 2.6), and how both knowledge graphs and social network features can aid fake news detection (presented in Section 2.7). Finally, we provide concluding remarks for this Chapter in Section 2.8.

2.1 Fake News, in the era of Social Networks

Information is vital in everyone's day-to-day life. Taylor [177] stated that information needs are both conscious and unconscious for any individual or group of people. There are also multiple types of information, such as knowledge, emergency announcements, and what has happened around us. Among all the information that aim to states facts, some of them are indeed factual, but some are non-factual. In this section, we first define fake news as used in this thesis, and introduce fake news in the era of social networks

2.1.1 Fake News Taxonomy

Non-factual information can also be classified into multiple categories based on their authenticity, intention, and if they can be strictly classify as news [207]. Specifically, Zhou and Zafarani [207] classified what commonly known as fake news into 8 categories as follows:

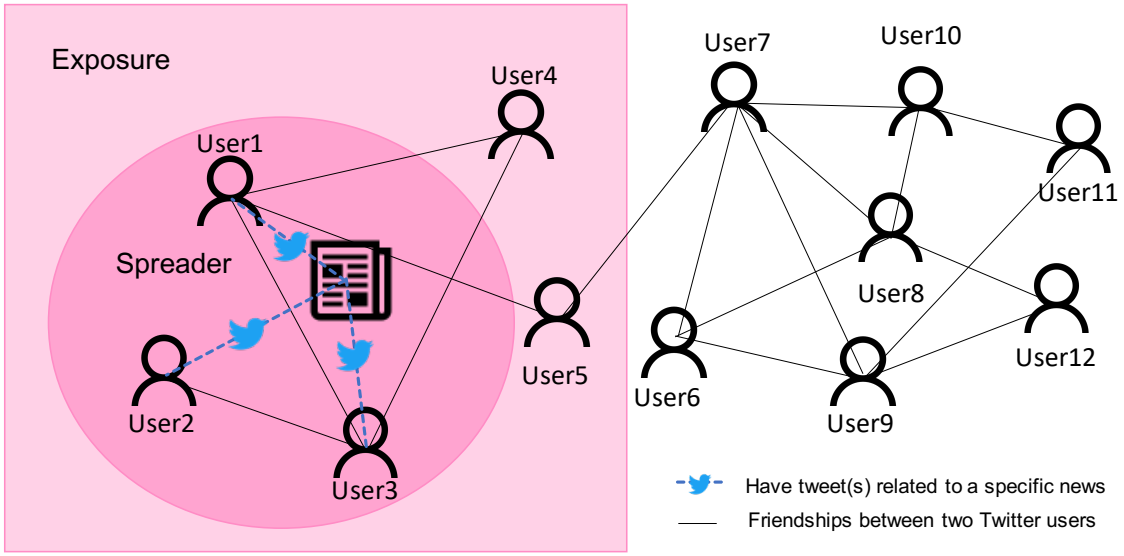


Figure 2.1: An example showing that users are connected on the Twitter platform, while only a subset of users engage with a specific news article.

1. **Rumour** is defined as random information that is presented in any format, with no clear intention, and is not necessarily false.

2. **Clickbait** is defined as information (article titles/article/sentences) that aims to mislead the public to increase its popularity, but not necessarily false.

3. **Cherry-picking** is defined as selectively reporting information with the intention of misleading the general public, where the information can be either factual or non-factual.

4. **Satire news** is a form of non-factual news articles/claims that aims to be entertaining.

5. **Misinformation** refers to information (news, opinion pieces, personal claims, etc) that is non-factual with unclear intention.

6. **Disinformation** is information that is non-factual with clear misleading intention.

7. **False news** is a generic term used to describe news that is non-factual, regardless of its intention.

8. **Deceptive news** is news with non-factual information with the intention to mislead the general public.

In this thesis, however, we do not distinguish these types of non-factual information, as intentions can be subjective, and hard to classify. Thus, in this paper, we consider all non-factual information, regardless of its intention or presentation, as fake news, and thus we aim to detect all the non-factual information online.

Next, we present the effects that social media has on the creation and spreading of fake news.

2.1.2 The Rise of Social Networks and their Roles in Spreading Fake News

After the broad adoption of the internet worldwide, knowledge bases, blogging, and social media networks are enabling faster and easier access to news than ever before, as well as the means to create and share any information [14]. Apart from reliable news agencies, independent reporters, unknown individuals, and even agents using fake identities can now build news websites to produce and spread information online. Moreover, social media outlets are becoming increasingly important in everyday life, where users can stay anonymous, obtain the latest news and updates, share links to news and information they want to spread, post and comment their own opinions, and use hashtags to make their opinions appear short and catchy. The sheer amount of information may create difficulties for information consumers to distinguish fake news from factual news. Indeed, users may not necessarily be aware that the information they encounter is false, and they may not have the time or effort to fact-check all the news and information they encounter online. Furthermore, misinformation online can render a large amount of information untrustworthy in the public eyes, can easily confuse and mislead the general public, and can cause public distrust in journalism [88, 173]. Unfortunately, such a reality is already unfolding, as news outlets and media reputation are increasingly in doubt amid a global distrust crisis [90].

For example, polarisation in political and scientific debates is observed more frequently than in pre-2007 [140, 142], since fake news can be amplified and reinforced through repeated exposure [135, 141]. Cho et al. [28] showed that one possible reason for the increasing polarisation is that the selected contents delivered by search engines (e.g., Google, YouTube) and social media platforms (e.g., Facebook, Twitter) are individually tailored to the users' specific interests, thus creating a *filter bubble* that reinforces their existing beliefs. Such an observation echoes the findings of Ling [102], who showed that individuals' false beliefs are emphasised by repeated exposure and a selective focus because of the *confirmation bias*. Indeed, the confirmation bias effect states that individuals are more inclined to read and interact with information that aligns with or confirms their existing beliefs, while they tend to avoid confrontational information and sources that challenge their existing beliefs [193, 204]. Furthermore, Yoo [199] identified *echo chamber* effects in social media platforms and showed that the echo chamber effects facilitate rumours and fake news being created and circulated within specific groups before being more widely spread. Echo chamber effects in social media can be described as a subset of like-minded users grouping together, with little interaction with dissimilarly-minded users, to preserve their beliefs and avoid confrontation [199]. For

example, Figure 2.1 illustrates that a news article might only reach a subset of users, with it being circulated and discussed by like-minded group members.

However, Lewandowsky et al. [94] argued that the misinformation crisis should not be recognised solely as a failure of individual judgements. Instead, they argued that it should be considered and evaluated as a public concern, especially with the popularity of social media.

Thus, journalists and scholars has been developing and researching methods in fighting the effects of rapidly spreading fake news. In the next section, we introduce the common practices among journalists in identifying fake news.

In the following sections, we introduce a set of methods that researchers have developed over the years to help detect fake news.

2.2 Machine Learning in Fake News Detection

In this section, we provide an overview of the general machine learning approaches regarding fake news detection. To identify fake news more effectively, researchers have been studying various ways to leverage the advances of machine learning (ML) models. ML models that aim to detect fake news online, usually use features extracted from news to classify if a news article is fake or factual. Some researchers have focused on using textual features from news articles, statements, and tweets; while some also focused on using social network information to detect fake news accurately. Regardless of the input type, there are generally four types of machine learning models that are widely used to detect fake news effectively.

2.2.1 Classic Statistical ML Models

Classic Statistical ML Models include a range of models that use statistical calculations to classify a given input. For example, the **Naive Bayes**(NB) classifier (e.g., [133]) is a probabilistic based model that applies Bayes' theorem on the input numeric features, to identify the most likely outcome in a finite given set; the **support vector machine** (SVM) classifier (e.g., [200]) uses Vapnik–Chervonenkis theory to separate samples into multiple categories, by maximising the width between any two categories; a **decision tree** (DT) aims to find the decision process where each numeric feature decides the possible route that can reach a conclusion; whereas the **random forest** (RF) model consists of multiple DTs, to be able to incorporate more features and create more complex routes to decide the possible classification results.

2.2.2 Deep Learning Models

Deep learning models, such as neural networks, are being recognised as an effective method for fake news detection [152]. Among all the deep learning methods, **recurrent neural networks** (RNN) (i.e., simple RNN, GRU, LSTM) consider time series when classifying the news, as it can be important to track changes and the emerging fake news. For example, Ma et al., [108] trained a multi-layer GRU based on the time series of the tweets, with a 5000 dimension TF.IDF score as input from each tweet to eventually predict an event's genuineness. This method yields a 10% performance increase in accuracy compared to non-deep learning methods (e.g., DT ranking, SVM, RF classification). Similarly, convolutional neural network (CNN) models have also been deployed in detecting fake news. For example Wang [188] proposed using a CNN model to analyse textual metadata from news articles, to identify non-factual news articles. Fang et al. [50] proposed to use self attention-based CNN based on the news articles' content, which outperformed RNN-based models on the identification of non-factual articles detection task. The machine learning approach is usually used with features extracted based on linguistic methods and static network analysis, where the most used features are introduced in Sections 2.6 and 2.7, but it lacks usage of dynamic network information.

2.2.3 Ensemble Models

Ensemble models aim to leverage the benefits from more than one model, where different models can use different architectures or different sets of features. For example, Reddy et al. [145] showed that an ensemble of logistic regression, random forest and Adaboost models trained on textual features outperforms each of the individual models, on the task of identifying non-factual articles; Liu et al. [105] showed that an ensemble of 25 differently tuned BERT models outperforms individual BERT, in identifying whether two news titles agree with each other; Das et al. [37] showed that a heuristic post-process on a selection of trained models achieved the state-of-art performance in identifying COVID-19 misinformation; Saeed et al. [153] showed that a conventional majority voting ensemble classifier fitted with three base learners, can enrich the traditional ensemble learner with deep contextual semantics from n-gram- based features and a convolutional neural network.

2.2.4 Propagation Models

Propagation models aim to use the propagation pattern of fake news on social media platforms to identify fake news on Twitter. Specifically, replying, liking, and retweeting are all ways to propagate a tweet to a larger audience on social media. Researchers found that the

propagation patterns can enrich the features of news spread on social media, and thus can facilitate the fake news detection task. For example, BranchLSTM [84] proposed to use the sequential manner of LSTM to analyse tweets threads (tweets, retweets, comments), in order to identify fake news contained in the tweet threads more accurately; Ma et al. [109] proposed to use kernel learning to study the propagation structures of microblogs platforms(i.e., Twitter and Weibo) to detect rumuors online.

2.2.5 Summary

This section introduced four types of machine learning methods for online fake news detection. Specifically, Classic machine learning models (e.g., NB, SVM, DT, RF) focus on using classic statistical models to predict if given news is fake or factual; Deep learning models (e.g., MLP, CNN, RNN) aim to classify news as factual or not using neural network architectures; ensemble models aim to capture the advantages of multiple models; propagation models focus on the propagation pattern of news online. In this thesis, we conduct experiments using statistical machine learning models, deep learning models, and ensemble models to find the best suitable model for each task. However, we do not focus on the propagation pattern of the news on social media platforms. Thus we do not conduct experiments using the propagation model.

2.3 Automatic and Human-in-the-Loop Fake News Detection Models

We have already introduced machine learning models that are widely used in detecting fake news in Section 2.2. In this section, we discuss two types of end-to-end systems design that focus on identifying non-factual news: automatic systems and human-in-the-loop systems. In the following, we describe these two types of systems used both on social media platforms and on news articles.

2.3.1 Automatic Fact-Checking Systems

To maximise the automation of the fact-checking process, automatic fact-checking systems aim to replace journalists in detecting fake news, and label news as fake or not directly by the systems, without the need for human journalists. For example, *FakeNewsTracker* [161] collected tweets that are associated with existing fake news proven by Politifact [1] and Buz-

[1] https://politifact.org

zFeed News [2], and extracted useful textual features to build machine learning models for fake news detection; *FAKEDETECTOR* [201] built a deep diffusive network to represent news articles, creators, and subjects simultaneously, based on a set of explicit and latent features extracted from the textual information; *ACT* [3] proposed to represent an article using a two-dimensional matrix that combines the aggregated credibility of the claim-article pair and the textual features from language models, and classify whether the article contains non-factual information using the two-dimensional matrix; *WikiCheck* [179] proposed to perform fact-checking using the Wikipedia Knowledge base by uncovering evidence that supports or refutes claims; *WebChecker* [181] used a reinforcement learning-based optimiser to find optimal checking plans and leverages various cost-accuracy tradeoffs to efficiently index, filter, and match news against existing fake news.

2.3.2 Human-in-the-Loop Fact-Checking Systems

Acknowledging the difficulties in building a fully automatic fact-checking system, human-in-the-loop systems employ journalists to manually check news with information automatically extracted by machine learning models. For example, *Scrutinizer* [77] reduced the manual fact-checking time by automatically classifying and highlighting the elements of the claims to be checked, and organised the specific questions that require human editors to clarify; *ClaimPortal* [112] provided an integrated web platform where journalists can target a specific claim on Twitter, and the models trained on a debunked claims database can help classify such claim as factual or not; *FactCatch* [127] proposed to first identify a set of claims it deemed valuable to be fact-checked from a claims poll and send the valuable claims to human fact-checkers, then a final judgement for the claim is calculated using the automatically inferred the truthfulness of the claim and the input of the human judges; *CoVerifi* [85] provided a platform to combine the classification results from fact-checkers and a GPT-2 model to identify fake news related to the COVID-19 pandemic. The GPT-2 model is trained with CoAID dataset (a recently constructed COVID-19 dataset [32]) to identify machine-generated text, and classify if claims are fake or factual. *Watch' n' Check* [22] proposed a platform that provides a keyword-filtering process where journalists can monitor and follow the discussion of a specific topic; *ClaimHunter* [12] adopted a reinforcement learning strategy, where the system sends claims and the machine generated predictions to journalists, and also uses the journalists' final verdicts to improve the system; finally, *WhistleBlower* [143] proposed to detect fake news using textual features generated by language models, but also allows fact-checking community members to change the verdict of fake news labelling using block-chain nodes.

[2] `https://www.buzzfeednews.com/topic/fake-news`

2.3.3 Summary

Recent advances in developing end-to-end fake news detection systems are plentiful. They can be classified into automatic fake news detection systems and human-in-the-loop systems. We acknowledge that human-in-the-loop systems can benefit from human input, but argue that such a system would still require a large amount of human labour. Thus, this thesis describes a fully automatic end-to-end fake news detection system based on our proposed FNDF. In the next section, we survey research related to the three proposed tasks presented in the thesis statement in Section 1.3.

2.4 Task-by-Task Fake News Identification

As presented in Section 1.3, we propose to tackle the fact-checking task using a two-phased and three-task framework. That is, we propose to identify fake news online by tackling the three following tasks: (1) identifying check-worthy sentences and tweets; (2) identifying if the check-worthy sentences and tweets are resurfacing fake news; and (3) fact-checking the sentences and tweets using social media platform. Next, we survey related work with regard to each task.

2.4.1 Task 1: Are these Check-Worthy?

The ClaimBuster system [71] was the first work to target the assessment of the check-worthiness of sentences. It was trained on data that was manually labelled as non-factual, unimportant factual, or check-worthy factual, and deployed SVM classifiers with features such as sentiment, TF.IDF, POS, and named entity linking (NEL). Focusing on debates from the US 2016 Presidential Campaign, Gencheva et al. [56] used many features from ClaimBuster, and found that if a sentence is interrupted by one participant in the middle of a long speech, it was more likely to be selected as check-worthy by at least one news organisation. There are many follow-up works [74, 134, 183] that have focused on deploying different learning strategies (e.g. SVM with various features, neural networks) in mimicking the check-worthy sentences selection process of a news organisation.

The CLEF'2019 CheckThat! Labs [6] provided check-worthy sentences from the 2016 US presidential debate, labelled by `factcheck.org`. Participants deployed learning models ranging from neural networks (LSTM [41, 69], feed-forward neural network [52]), to traditional methods (i.e. SVM [168], naïve Bayes [31], logistic regression [41], regression trees [4]), with features ranging from embeddings (i.e. word [41, 54], part-of-speech tagging [41], syntactic dependence [69], Standard Universal Sentence Encoder [52]), BoW

related (i.e. TF.IDF [31, 54, 168], n-grams [4], named entities [4, 54], POS [4, 54]), sentiment [52, 54], topics [4], to readability [52], and sentence context [52]. However, none of the teams using language models and handcrafted semantic models achieved a mean average precision higher than 0.19, suggesting that the task is challenging, and that the language models and handcrafted features can not yield satisfactory results (**Gap 1**).

Moreover, the CLEF'2020 ChechThat! label [11] expanded on the check-worthy task to include the task of identifying check-worthy tweets of a given topic. Similar to the check-worthy sentence detection, participants deployed a range of machine learning models (e.g., RF [115], SVM [24],CNN [2]) using textual features (e.g., TF.IDF [115], part-of-speech tagging [24, 78],and named entity tagging [24]). Deep language models, such as the BiLSTM [114], pre-trained BERT [2, 24, 78] and RoBERTa [131, 132] are also deployed in identifying check-worthy tweets. Among all the participating models, team Accenture [132] achieved the highest mean average precision score of 0.806, by representing tweets use the pre-trained RoBERTa model.

Identifying entities within sentences to be checked has been used in the suspicious claim identification literature. For instance, Altun et al. [4] and Gąsior et al. [54] used named entity recognition to identify the types of entities present in a sentence (e.g. person, location, organisation, money). However, such methods do not account for the rich information an entity contains, which can be combined with the language model representations (**Gap 2**). In contrast, this thesis proposes to use recent advances in dense knowledge graph embeddings [16, 100, 129, 180, 194, 195] to provide rich information for suspicious claim identification. We hypothesise that by integrating entity embeddings with language model representations, we can improve the performance of our selected language model in identifying check-worthy sentences and tweets. The following section surveys the studies that focused on detecting recurring fake news online.

2.4.2 Task 2: Identifying Recurring Fake News

Given the identified check-worthy tweets and claims, we focus on classifying whether the identified check-worthy news is indeed non-factual. The WSDM Cup 2019 fake news challenge addresses the task of matching new articles with previously identified fake news to identify **recurring** fake news. In this challenge, the winning group *saigonapps* [139] used a BERT-based language representation and handcrafted features to represent each title pair; these features were then ensembled to produce the final result, obtaining 88.2% accuracy; team *Travel* [105] ensembled twenty-five BERT models with six other models to produce the final verdict on if the news title matches the existing non-factual news; and finally team IKM lab [196] combined dense RNN and CNN as an ensemble architecture to identify if the news title aligns with the existing fake news.

The task of identifying recurring fake news is not heavily investigated in other challenges or studies, and remains an open research question (**Gap 3**). Nevertheless, we argue that recognising if newly emerged fake news are in fact recurrent fake news can be beneficial, given that fake news can resurface multiple times online [150, 159]. Thus, we consider it an essential component for an online automatic fake news detection framework, and incorporate it into our proposed end-to-end automatic fake news detection framework.

2.4.3 Task 3: Fake News Detection on Twitter

Twitter is a prominent social network platform where users around the world can obtain recently publish news and share their thoughts with anyone online. Researchers have used Twitter to investigate the propagation pattern of fake news, and developed models to use matadata from tweets and users in fake news detection tasks. For example, Twitter can provide static information based on a snapshot of the current connections of a specific user, where each user is treated as a simple node with multiple numeral features unrelated to the user's characteristics. Some numerical features used in fake news detection task from users and tweets are as follows: simple user features (e.g., # of followers, verified or not, description) [97]; relations between users (e.g., followers/followees, in the same region, engaged with the same tweet/URL) [152]; and relations between tweets (e.g., replies, retweets, likes, viewpoints conflicts) [75].

As an example of using relationships between tweets, Jin et al., [75] showed that using the credibility of each tweet calculated through a tweet-tweet relation matrix, their model can outperform previous models using textual features alone. However, all the methods mentioned above that include user information as one of the features only consider statistical information about the users whereas how to best use the rich network information social networks can provide remains an open research question (**Gap 4**).

2.4.4 Summary

This section summarised studies that focused on each task in detecting fake news in our proposed FNDF. As a result, we identify the following general gaps in this section:

Gap 1: Language models and handcrafted features generally perform non-satisfactory, which are the most common types of features used in detecting fake news online.

Gap 2: Sophisticated language representations are well used in identifying check-worthy tweets, but there is limited work on whether they can be combined with entity information in the existing literature.

Gap 3: The identification of recurring fake news is not well studied and remains an open research question.

Gap 4: How to effectively use the dynamic social media users' connections with each other in detecting fake news is still an open research area.

Combining the above mentioned four gaps, we identify **Gap 5**: The existing end-to-end fake news detection systems cited in Section 2.3 largely did not consider the filtering process, when not all tweets and sentences require fact-checking; they mostly overlooked the recurring fake news detection process; and generally did not leverage the user network structure to analyse the users engaging in fake news in a graph analysis manner.

This thesis proposes to address **Gap 1** by experimenting with a range of large language models, to identify the most effective language model for the specific task. In Chapter 4, we propose to address **Gap 2** by investigating how to best represent entities within the text, to more accurately identify check-worthy claims and tweets. In Chapter 5, we propose to address **Gap 3** by identifying the best approach in identifying recurring fake news using an existing dataset. Chapter 6 proposes to address **Gap 4** by using user network analysis to identify fake news. We present our proposed framework in detail in Chapter 3. Finally, Chapter 7 combines all the components we proposed in the thesis, and addresses **Gap 5**. We also elicit **Gap 1**, **Gap 2**, and **Gap 4** in the following sections.

In the following sections, we introduce a list of fake news datasets that journalists and researchers have published in related to fake news detection.

2.5 Datasets for Fake News Detection

Journalists and scholars have published a range of fake news datasets, allowing the wider research communities to analyse common features of fake news and develop tools and models to help combat them. Thus, this section focuses on introducing a list of datasets that are made publicly available.

- *ClaimBuster* [70] contains 30 presidential debates between 1960 and 2012. In total, c sentences spoken by the presidential candidates are labelled as Non-Factual Sentences, Unimportant Factual Sentences, and Check-worthy Factual Sentences.
- *ChechThat!* [6, 34, 124] is a annual challenge, where the 2019 and 2020 datasets consist of transcripts of US political debates and speeches in the time period 2016-2019, collected from various news outlets[3]. The organisers manually compare each sentence with `factcheck.org`. If the sentence appeared in `factcheck.org` and is being

[3] ABC, Washington Post, CSPAN, etc. [11] are in English only.

fact-checked, it is labelled as a check-worthy claim. The CLEF'2021 Task 1a English dataset consists of tweets collected concerning COVID-19 and manually identified as either check-worthy or not check-worthy.

- *BuzzFeedNews* [4] focuses primarily on the 2016 U.S. election. It contains all the news published from 19th to 23rd, and 26th to 27th September 2021, from 9 news agencies (ABC news, Addicting Info, CNN, Eagle Rising, Freedom Daily, Occupy Democrats, Politico, Right Wing News, and The Other 98%). In total, it contains 1627 news articles, where 826 are from mainstream news sources, 356 from left-wing news sources, and 545 from right-wing news sources. Furthermore, all articles are fact-checked by five journalists from BuzzFeed. Following this dataset, BuzzFeed News published two more datasets related to fake news in the following years, namely *Top 50 fake news of 2017* and *Top 50 fake news of 2018*.

- *BuzzFace* [154] extends the BuzzFeed dataset by collecting the comments related to news articles on Facebook. The dataset contains 2263 news articles and 1.6 million comments discussing news content.

- *FacebookHoax* [175] also contains posts published on Facebook. Specifically, FacebookHoax comprises real scientific news (from scientific pages) and conspiracy hoax (from conspiracy pages). It contains 15,500 posts on 32 pages, where 14 pages are conspiracy pages and 18 are scientific.

- *CREDBANK* [120] is a large-scale crowd-sourced dataset that contains around 60 million tweets between 10th October 2014 and 26th February 2015. This dataset contains more than 1300 events, with credibility ratings and reasons collected via Amazon Mechanical Turk.

- *PHEME* [210] contains tweets and their replies/retweets of five breaking news and four specific known rumours. We include all the 2,695 rumourous source tweets in our existing fake news collection.

- *CoAID* [32] dataset contains claims, news articles, and engaged tweets that are labelled as fake and not fake. We include the titles of debunked fake news, fake claims and fake tweets in our existing fake news collection, which amount to 498 debunked fake news.

- *FAKENEWSNET* [162] contains gossip and political fake news gathered from *PolitiFact*[5], and *GossipCop*[6]. The dataset contains news articles labelled as fake or real. The FAKENEWSNET dataset also contains Twitter engagement of collected news by applying Twitter search on the news article title.

[4] https://github.com/BuzzFeedNews/2016-10-facebook-fact-check
[5] https://www.politifact.com/
[6] https://www.gossipcop.com/

- *LIAR* [188] also contains claims gathered from *PolitiFact*. The Liar dataset differs from the *FAKENEWSNET* dataset because they did not collect the news stories being judged as fake or real. Rather, *LIAR* gathered the statements made in political speeches and debates. The statements are labelled as "pants-fire", "false", "barely true", "half-true", "mostly-true", and "true".

- *the WSDM 2019 Cup Fake News Challenge dataset*[7] consists of human-written Chinese news title pairs, that are labelled either *unrelated*, *agree*, or *disagree* with a given debunked fake news.

- *MM-COVID* [96] consists of 2492 non-factual and 5311 factual source contents (news and tweets) related to COVID-19 that are labelled as fake or not, and related tweets that have engaged with the source content.

- *SD dataset* [128] is a fake news dataset focusing on the news being shared on Twitter. It consists of news article links and human judgements labels denoting if they are fake or not, as well as engaged tweets, the stance of such tweets, the publisher of the news article, and article citations by other news outlets on Twitter.

The datasets mentioned above contain different types of information and features in fake news detection. Specifically, the ClaimBuster dataset concerns claim check-worthy but do not distinguish between check-worthy false and non-check-worthy false claims. The CheckThat! Challenge datasets provide check-worthiness labels for both U.S. presidential debates and news related to COVID-19 but lack the factual label. BuzzFeedNews focuses on news articles from a few selected established news publishers. Buzzface contains both news articles and user comments, but does not capture temporal information. The FacebookHoax dataset contains many Facebook posts from a few instances of pages, while all posts from conspiracy pages are labelled hoaxes, thus may be biased. The CREDBANK dataset contains both tweets and events, but tweets and events are separated and not correlated. PHEME contains only fake claims related to five events, making it a narrow-focused fake news dataset. CoAID contains only fake news on COVID-19, making it narrow-focused. Although FAKENEWSNET contains news articles, related tweets, and other Twitter engagement, they focus on identifying the entire article as fake and do not focus on claim/sentence/tweet level factualness. Finally, the LIAR dataset contains only short claims with Politifact verdict, without social media engagements. The WSDM 2019 Cup Fake News Challenge dataset contains news title matching but does not provide any social media engagement. MM-COVID dataset focuses on tweets only related to the COVID-19 topics and thus is narrowly focused. The FANG dataset contains only 1054 fake and real news with Twitter engagement.

[7] `https://kaggle.com/c/fake-news-pair-classification-challenge/data`

None of those mentioned above datasets satisfies all our requirements of this thesis: claim/sentence/tweet level check-worthiness labels, claim/sentence/tweet level truthfulness labels, and social media engagement information. Thus, this thesis uses different datasets to evaluate individual tasks, and we use the SD and MM-COVID datasets for our final end-to-end evaluation.

The following section surveys recent studies that focus on extracting textual information from fake news online, which are related to the methods we propose to address **Gap 1**.

2.6 Language Models Advances

This section focuses on the existing studies that aim to analyse and understand textual features in articles, tweets, or speech transcripts. We describe the methods commonly used in text analysis in two categories: traditional textual analysis and neural language models. We first discuss the traditional text analysis methods.

2.6.1 Traditional Text Representations

2.6.1.1 Words Occurrences Analysis

The commonly used methods to represent text in machine learning algorithms include the bag-of-words (BOW) (e.g., TF.IDF) [76] and the part-of-speech (POS) [20] approaches. The simplest text representation method, BOW, focuses on the words' occurrence and frequencies only, while POS tags are based on lexical cues (i.e., noun, verb, location, time, etc).

On the other hand, tweets are a particular type of text since they are short and contain hashtags and links. Thus, scholars usually combine the TF.IDF/POS methods with other occurrence-based linguistic methods and statistical information of tweets especially designed for tweet content. Some examples are the number of mentions of other users (@), the number of hashtags (#), the time-span of the tweet's reply, the tweet location, etc [67].

2.6.1.2 Syntax Analysis

The syntax of text [79], usually viewed as grammar, is also helpful in detecting fake news. This method transforms text into a syntax tree, such that the grammatical information between nouns and verbs, subjects and objects, are revealed and used for further analysis. For example, Feng et al. [53] showed that using features driven by Context Free Grammar parse trees can achieve 70%-90% accuracy on the deception detection task over four datasets.

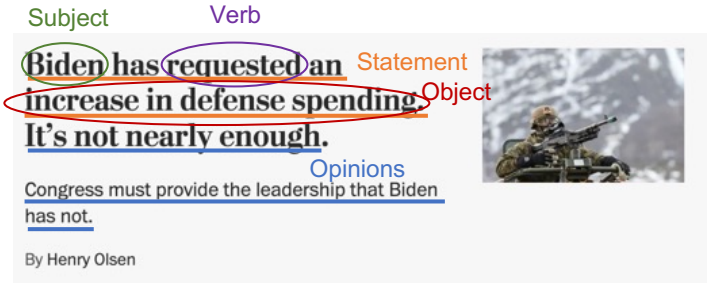


Figure 2.2: An example of a news article title.

However, Figure 2.2 shows the title of a news article, which has the subject, verb and object of a statement, and one opinion-based commentary; while the secondary title is an additional opinion-based commentary. In this case, the syntax representation methods mentioned above cannot identify the commentaries from the statement, due to a lack of deep understanding of the text. Therefore, using syntax analysis alone may not be sufficient in detecting fake news.

2.6.1.3 Semantic Analysis

Semantic analysis produces language-independent meanings of a given text and extends both word occurrences analysis and syntax analysis. As a result, scholars are able to understand the underlying information and meanings of a given text using semantic analysis. There are two types of semantic analysis methods often used in fake news detection, as it shows the subjective motivation of the authors:

1. **Sentiment Analysis.** Sentiment analysis aims to answer whether a piece of text expresses positive, neutral, or negative sentiment. It is helpful to understand the sentimental opinion, using a single assignment of the sentiment score (usually ranges from -1 to 1) or a label (i.e., positive, negative, neutral). In fake news detection tasks, sentiment is shown to be helpful. For example, Hamidian et al. [66] showed that sentiment analysis improved the model performance on the fake news detection task. Moreover, Ghenai and Mejova [57] used information gain (IG) with the greedy backward elimination method to find the most valuable features in predicting whether a tweet contains rumours about the Zika virus. They showed that sentiment score is the fourth-best feature according to IG.

2. **Emotion Tagging.** Vosoughi et al. [186] found that false rumours-inspired tweets express emotions very differently compared to truth-inspired tweets. Specifically, surprise and disgust are expressed more often in false rumours, while sadness, anticipation, joy, and trust are expressed more often in truth-related tweets. This observation shows that users display different emotions toward non-factual and factual news, but

whether it is because truthful and fake news affect users' emotions differently, or because user groups often express different emotions are gathering around different types of news, remains unclear. Either way, emotion appears to be a good feature in identifying fake news. Furthermore, Vosoughi et al. [186] also found that false rumours contain significantly more novel information than factual news, when compared using the Latent Dirichlet Allocation (LDA) topics between false rumours to the tweets the user was exposed to in the previous 60 days.

2.6.2 Neural Language Models

Neural language models have become the most widely used language representation methods in recent years. Neural language models aim to analyse words' context better, using the advances in computational power and practical applications of neural networks. The basic ideas of using neural language models to represent text are threefold:

1. **Representing tokens in a lower-dimensional space.** Neural language models aim to represent a token in a finite-dimensional space rather than encode each token with a unique id. As a result, the tokens with similar meanings are closer to one another in such lower-dimensional space.

2. **Representing tokens as vectors.** To effectively represent tokens in the lower dimensional space, neural language models often use real number vectors that represent the location of the token in this space, where the dimension of the vectors equals the dimension of the lower-dimensional space.

3. **Inferring the embedding of a token based on its surrounding or preceding tokens.** The semantic meaning of a token is not isolated from its surrounding tokens. Thus, neural language models generally aim to pull tokens closer if they occur together frequently or have similar semantic meanings.

For example, in 2013, Mikolov [117] proposed Word2Vec, a shallow neural language model trained based on the co-occurrence of words in a text, which aims to encapsulate the semantic meaning of each word, and thus can help machines to better encode the semantic meaning of the sentences in a single vector. Similarly, the Doc2Vec model [91] embeds documents using the Word2Vec method, but focuses more on the semantical and contextual information within each document.

Proposed in 1997, but not widely used until after the popularity of Word2Vec, the RNN based language models (e.g., the long short term memory model [73] (LSTM)) can be applied to analyse and process textual features. For example, Ma et al. [108] used LSTM networks

to represent text sequentially, capturing the semantic meaning of the text based on previous tokens. LSTMs differ from Word2Vec as the LSTM models encapsulate information of the long previous sequential tokens, whereas Word2Vec representations are usually based on immediate surrounding tokens of the token being embedded. However, RNN models do not consider the future context when predicting language. To address this disadvantage, researchers have developed bidirectional RNNs [157] (e.g., BiRNN, BiLSTM) that capture both previous and future tokens in a sentence.

More recently, attention-based [184] neural network models (e.g. ELMO [137], BERT [40], ALBERT [89], RoBERTa [106], BERTweet [126]) use the attention mechanism to identify relevant contexts within or between sentences. These attention-based language models combine the advantages of an extensive complex neural network with their pretraining on a large corpus, to create pre-trained language models (e.g. BERT is trained on Wikipedia, BookCorpus, and Common Crawl [40]), where the subjective bias from any small training data is minimised. Being one of the state-of-the-art pre-trained language models, BERT is shown to consistently outperform other shallow language models and (Bi)RNN-based language models in many tasks (e.g., reading comprehension [182], document retrieval [172, 198], question answering [107]), thanks to its large pretraining data, as well as the flexible representation of words, based on their surrounding context.

2.6.3 Language Models in Fake News Identification

Textual analysis is the focus of detecting fake news in many studies [56, 71, 75, 139]. However, the existing studies [2, 24, 78, 114, 131, 132] that applied deep neural network language models to the task of identifying check-worthiness detection did not compare the performance across the popular pre-trained language models to identify the most suitable language models for the check-worthiness task (**Limitation L1**). Furthermore, the above mentioned studies mostly only use language models as the sole embedding features, without considering other types of information, such as entity information in the text (**Limitation L2**). Thus, how to best leverage language models, and how to incorporate other non-semantic information with the semantic-focused language models remain open questions.

2.6.4 Summary

Whether delivered as a news article or as a short statement on Twitter, news typically consists of text – an essential information carrier. Analysing the text is unavoidable in identifying fake news in articles, debate transcripts, or social media platforms. Recent advances in text analysis allow us to analyse text using more sophisticated models. However, on eliciting **Gap 1** identified in Section 2.4.4, we identify the following limitations:

Limitation L1: It's unclear which deep learning language model is the most suitable for identifying check-worthy sentences/claims/tweets, and detecting recurring fake news.

Limitation L2: The language models introduced above mainly focus on only the semantic features, whereas the other features - such as entity-related information - are not considered.

In this thesis, we deploy language models to analyse text and modify some of the models so they are more tailored and suitable for the tasks at hand. Specifically, we propose the following to enhance the performance of the language models used in our framework to detect fake news.

1. We propose to identify the best language model to identify fake news on Twitter, to address **Limitation L1**.
2. We propose to combine traditional textual features with neural language models to identify the entailment of two news titles effectively, to address **Limitation L2**;
3. We propose to combine neural language models with named entity information to identify news that needs further fact-checking, to address **Limitation L2**;

In this thesis, we study the effectiveness of various language models in identifying check-worthy claims and tweets (Chapter 4) and identifying recurring fake news (Chapter 5). We also compare the effectiveness of using textual features with using network features in identifying fake news on Twitter (Chapter 6). However, although textual features are essential in fake news detection, they may not be sufficient as the only type of information needed in detecting fake news online. Thus, in the next section, we introduce networks-based (i.e., knowledge graph and social networks) features as additional information that can aid the task of detecting fake news.

2.7 The Rise of Graph Embeddings

Networks, usually represented as a linked graph, have data points linked to other data points through relationships. Knowledge graphs and social networks are the two main types of networks that are particularly rich in conveying additional information that textual features may not capture when detecting fake news online [163, 206]. To effectively use network information in fake news detection, this section describes the recent advances in methods that aim to embed both knowledge graphs (in Section 2.7.1) and social networks (in Section 2.7.2).

2.7.1 Knowledge Graph Embeddings

A knowledge base (KB) usually contains entities (nodes) and connections (edges) between two entities, where there are usually finite types of relationships (different types of edges). Each edge in a KG is usually represented by a triplet $\langle e_h, r, e_t \rangle$, indicating that the head entity e_h and tail entity e_t are connected by relation r, e.g., $\langle$*Donald_Trump*, *NomineeOf*, *United_States_presidential_election_2016*$\rangle$. Representing a KG in triplets is effective to convey factual and trackable relationships, and thus can facilitate the fact-checking processes (e.g., [99]). However, a KG is relatively hard to represent in a lower-dimensional vector space, because of the complex types of nodes and edges associated with it. There are many existing approaches [15, 16, 129, 189] that learn embeddings from KGs, by training neural network models based on the co-occurrence of entity pairs and relationships. Generally, two types of models are widely used to train KG embeddings: distance-based KG embeddings with "facts alone" models [16, 23, 189] trained on a semantic triplet graph alone (such as FB15k [16]), while semantic-based entity embeddings [15, 129] also use the information contained in the corresponding entity descriptions (e.g. Wikipedia pages). We describe these two types of models in turn below.

2.7.1.1 Facts Alone KG Embedding

Freebase[8] [13], Google Knowledge Graph[9], GeneOntology[10], and Wordnet [119] are widely used multi-relational knowledge graphs (i.e., contrary to single-relational KG, where there is only one type of relation that connects two entities), where the entities consist of abstract concepts and concrete entities of the world, and the relationships are the facts that link each pair of entities together. Derived from Freebase and WordNet respectively, FB15k [16] and WN18 [15] are two widely used *facts alone* knowledge bases in training KG embeddings, while wikidata [187] has become one of the most popular knowledge bases in recent years. The structure of a relation triplet (i.e., a triplet in the form of $\langle e_h, r, e_t \rangle$, without any additional descriptions for e_h, r, e_t) in such knowledge bases enables KG to represent information hierarchically and graphically. Moreover, such representation can be represented in a lower dimension space using graph embeddings, where the scoring functions are generally based on distances between entities and relationships.

For example, TransE [16] used the Euclidean distance between two connected entities (i.e., e_h and e_t) to project the entities and relationships into a learnt vector space, while TransR [100] projected the entities and relationships into different spaces (e.g., one space for the entities, one space for the relations). RESCAL [129] and DistMult [195] projected such distances

[8] `https://freebase.com`
[9] `https://google.com/insidesearch/features/search/knowledge.html`
[10] `https://geneontology.org`

into different vector spaces using tensor factorisation. The advances in deep neural networks also encouraged researchers to deploy deep neural networks on graph-structured data, such as data encapsulated in a KG. For example, Li and Madden [95] combined a graph embedding method *node2vec* [61] with a cascade embedding method, achieving better performance at predicting triplets than using node2vec method alone.

However, such methods did not consider the differences between a symmetrical triplet – where both entities within a triplet can be considered as either the head or the tail (an example of a symmetrical triplet is $\langle$ *Barack Obama, married to, Michelle Obama*$\rangle$, which can also be represented as $\langle$ *Michelle Obama, married to, Barack Obama*$\rangle$); and an asymmetrical triplet – where the triplet's entity head can not be viewed as the triplet's tail (e.g., $\langle$ *Stanley Kubrick, directed, Dr.Strangelove*$\rangle$ (asymmetrical) which **cannot** be represented as $\langle$ *Dr.Strangelove, directed, Stanley Kubrick*$\rangle$). To allows the binary relationship embeddings to represent both symmetrical and asymmetrical relationships, ComplEx [180], RotateE [174], QuatE [202] projected KG into a complex space represented with complex numbers, and thus distinguish the symmetrical triplets from asymmetrical triplets.

Finally, MuRP [10] and RotH [23] used the hyperbolic space (which consists of a constant negative curvature that can represent discrete trees in a continuous analogue) to model a KG. One entity's multiple possible hierarchical relationships can be modelled simultaneously in such a hyperbolic space, resulting in fewer dimensions of embeddings, thereby achieving better performances than those obtained by the Euclidean distance methods.

2.7.1.2 Semantic-based KG Embeddings

Some knowledge bases (e.g., DBpedia) contain more information than just triplets of entities and relationships (e.g. text descriptions for entities, relationships, and their possible features, such as $\langle$*Stanley Kubrick, directed, Dr.Strangelove, a comedy/war movie*$\rangle$)). Hence, a semantic analysis of the available descriptive texts allows algorithms to better capture each entity and its semantic meaning, where a hyperlink between entities serves as a relationship between the two linked entities. To this end, jointly training KG embeddings with semantic embeddings can benefit one another. For example, researchers have explored traditional machine learning methods on jointly trained embeddings, such as random walk [62]. He et al. [72] used deep neural networks to compute representations of entities and contexts of mentions from the KB, and Yamada et al. [194] used a skip-gram method, and trained it on Wikipedia data to obtain the entity embeddings and the associated word embeddings.

More recently, some researchers (e.g., ERNIE [203], KnowBERT [138]) have explored the use of joint training knowledge graph embeddings along with a BERT language model, and showed promising results in several downstream tasks. Bosselut et al. [17] explored whether using the attention mechanism (similar to that for training the BERT model) to enrich a

knowledge base embedding with "common sense knowledge" embedded in text content is beneficial for more complete KG embeddings. The resulting model, named Comet, puts more emphasis on general information represented as entities (e.g., ⟨nap, having sub-event, dosing off ⟩).

2.7.1.3 Knowledge Graph in Fake News Detection

Knowledge graphs are able to provide structured information for entities and relations, and are used for a wide range of downstream tasks, such as entity linking [26, 121], relation prediction [125, 192], and knowledge graph completion [101, 158]. Existing methods that use knowledge graphs in fake news detection largely focused on named entity linking [24, 71], constructing facts using Wikipedia [179], and similarities between entities [168]. However, research are limited on whether embedded entities are beneficial in detecting fake news (**Limitation G1**). Thus, how to tailor the entity representations for fake news detection (**Limitation G2**), and how to choose the best entity embedding models for fake news detection (**Limitation G3**) remain open questions.

2.7.1.4 Summary

This section surveyed types of knowledge graph embedding models that have been proposed in recent years. We elicit **Gap 2** into the following:

Limitation G1: Whether the embedded entities are beneficial in identifying check-worthy claims/tweets/sentences, and in detecting fake news has not yet been studied.

Limitation G2: The current entity embeddings obtained using knowledge graphs are not tailored toward fake news detection tasks. How to represent a pair of embedded entities so that they are the most beneficial in detecting check-worthy tweets and sentences/claims is still an open research question.

Limitation G3: It is not yet identified which entity embedding method is the best for identifying check-worthy tweets/sentences/claims.

This thesis proposes to address **Limitation G1-3** by conducting experiments that combine language models together with entity embeddings, on the check-worthy tweets/claims/sentences identification task. The detailed experiments are presented in Chapter 4. The next section focuses on surveying the recent research related to social network embedding models and their usages in fake news identification.

2.7.2 Social Network Embeddings

Social networks (such as Twitter) are platforms where people can connect with other people online, and share information and news with their followers and friends. The links between two users A and B on Twitter can be explicitly categorised into two types: *friends* or *followers*, that is, if user A follows user B, then A is B's **follower** and B is A's **followee**. If B also follows A, A and B are each other's **friends**. Researchers have developed a range of methods that embed the rich social networks' structure into a lower dimensional space, and some researchers also conducted experiments that used such network embeddings to analyse social media platforms. We now present the social network embedding models used in building embedded social networks.

2.7.2.1 Embedded Social Network Features

Recent advances in constructing graph embeddings most aimed to represent a graph in a lower-dimensional space, where each node (user) and/or vertex (user's connection) are represented as a vector-based on their neighbouring nodes (friends or followers) and the vertices (friendship or followership) that connect them.

Typically, graph embeddings are achieved by embedding only the topological structure of graphs [61, 136], or using both the topological structure and the auxiliary information of the graph, such as the content of nodes [27, 68]. For instance, the node2vec model [61] and the DeepWalk model [136] are the first two approaches developed to represent the graph in a lower-dimensional space, with deep learning techniques, based on topological graph information. These two methods are similar, as they both use a Skip-Gram architecture (that allows tokens to be skipped while forming adjacent n-grams) with negative sampling to learn the embeddings of each node within the graph, based on the portion of the graph generated by random walks/edge sampling.

However, such methods only consider the local context (i.e., closest neighbouring nodes) for a given node, and cannot obtain a global optimum in representing the complexity of a graph. To better represent a node within a complete graph structure, graph convolutional networks [82] (GCNs) were proposed, which aim to capture the global structure of a given graph. Typically, a GCN model uses several layers of graph convolution operations, where each layer is built to capture the information of each node's neighbours, which is then fed to the next layer, therefore achieving convolutional learning of the graph structure.

However, such GCN models have a very high computational cost [27], as the whole graph's structure and each embedding layer's information have to be stored. Thus, various methods have been proposed to reduce the consumption of computational time and space, such

as mini-batch stochastic gradient descent with variance reduction GCN [25], and Cluster-GCN [27]. For example, the GraphSAGE [68] method was developed to perform parameterised random walks and uses recurrent aggregators. It can be used for both unsupervised and supervised representation learning with a proximity loss between nodes. Moreover, the model adopts a dynamic inductive algorithm to generate embeddings for unseen nodes and edges at inference time.

Graph embedding methods have been shown to be effective in many tasks, such as node classification [136], link prediction [81, 151], and social networks alignment [45]. Embedding social network using graph embedding models are also used to detect fake news on social media platforms. For example, Röchert et al. [147] studied the user network structure of YouTube channels. They found that the channels and individuals propagating fake news on YouTube are usually integrated into heterogeneous discussion networks that involve factual content more than misinformation. Sosnkowski et al. [165] showed that changes in the users' network structure on Twitter can help detect the change in political opinions among users. The Factual News Graph model (FANG) of Nguyen et al. [128] proposed to use inductive learning for representing social structure, and combined the graphical social network information with sophisticated textual features. Specifically, Nguyen et al. created a citation network based on the news citations among publishers, and labelled user engagement with the news articles with stances (i.e., support, deny, comment (neutral) and comment (negative)). The additional information needed to construct such a user network in detecting fake news is labour intensive and time-consuming (**Limitation N1**). Similarly, Rath et al. [144] proposed to use the network structure information to identify the potential super spreaders of fake news, and showed that the proposed model can identify potential spreaders with the retweet network, the follower-following network, and the timeline data. However, such user network structure is not tested in detecting fake news, and the type of user network structure that is most effective in detecting fake news is yet to be identified (**Limitation N2**).

2.7.2.2 Summary

This section summarised the two groups of features widely used in social network embeddings are statistical user features and network features. The limitations that elicit from **Gap 5** are as follows:

Limiation N1: Current users embeddings for fake news detection are trained on complex networks that require additional labelled data. These networks are challenging to collect and thus impractical in training on a large dataset.

Limitation N2: The most effective type of users connections to use in constructing a user network in detecting fake news on Twitter is unknown.

This thesis proposes to use inductive representation learning of social network structures in a fake news detection task to tackle the disadvantages mentioned above. Specifically, to address **Limitation N1**, we aim to use the readily available data that can be obtained directly through Twitter - without further processing - to construct a user network embedding that can accurately cluster users into different groups, based on their friendships with other users and their followers. Building on the users' network embeddings, we then aim to represent a news story using the aggregation of the engaging users, to predict where the news story is fake. To address **Limitation N2**, we construct two user networks – a user follower network and a user friendship network, and compare the performances of using user embeddings obtained from either network structures, in detecting fake news. We present the experiments that address these two limitation in Chapter 6.

2.8 Conclusions

This chapter presented some previous research focusing on identifying fake news online. Section 2.1 first introduced the fake news phenomenon in the internet era, why it is dangerous, and how did the public react to them. We also introduced possible psychological reasons that draw some social media users into believing fake news online. Then Section 2.2 introduced the general practice in identifying fake news online through machine learning approaches. We introduced the two types of end-to-end systems for tackling fake news in Section 2.3. Section 2.4 introduced three tasks that can help tasking fake news, and surveyed related works in each task. We also identified the 5 general gaps (**Gaps 1-5**) that need to be addressed, to more accurately identify fake news. Section 2.6 focused on surveying the language models and textual features that are widely used in the fake news identification task, where we presented 2 limitations (**Limitations L1 and L2**). Section 2.7 presented related work in constructing and using both knowledge graphs and social networks in fake news detection. Specifically, Section 2.7.1 surveyed related works w.r.t. knowledge graph and entity embeddings, where we identified 3 limitations (**Limitations G1-G3**). Finally, Section 2.7.1.2 presented recent studies regarding social network embeddings. **Limitations N1 and N2** are identified in this section.

In the next chapter, we formally present our proposed framework, FNDF, to effectively identify fake news.

Chapter 3

Framework Overview

3.1 Introduction

In Section 1.3, we argued that the accurate identification of fake news online can be achieved with a two-phased framework. In the previous chapter, we provided a background review in the field of fake news detection. Furthermore, we elicited five gaps between the current techniques in detecting fake news and the new advances needed for tackling such a task. In this chapter, we introduce our proposed framework introduced in Section 1.3, that can bridge the five gaps layed out in Section 2.4.4.

In particular, as discussed in Section 2.4.4, **Gap 5** states that existing fake news detection systems have largely overlooked the process of identifying check-worthy tweets and sentences, so as to focus the limited computational power on identifying whether the check-worthy tweets and sentences are factual or not. They also failed to identify recurring fake news, although existing fake news collections can be used to identify resurfacing fake news. And finally, information from dynamic user networks are overlooked to leveraged the network structures in fake news detection. This chapter presents our Fake News Detection Framework (**FNDF**) that aims to address **Gap 5**. FNDF, consists of two phases and three tasks, aims to leverage multiple aspects of a sentence/tweet, existing fake news datasets, and user engagement on social media to identify fake news more effectively than the existing systems introduced in Section 2.3, which mainly rely on either linguistic features [3, 161, 181] or statistic network features [179, 201].

Specifically, Task 1 aims to address **Gap 2** by combining entity representations with textual representations, to effectively identify check-worthy sentences and tweets. Task 2 aims to address **Gap 3** by assessing the entailment between a check-worthy tweet or sentence with existing debunked fake news, to identify recurring fake news. And finally, Task 3 aims to address **Gap 4** by representing a check-worthy tweet or sentence using the embeddings of

its engaging users, to effectively identify fake news leveraging social network features.

We structure the remainder of this chapter as follows: Section 3.2 provides an overview of our proposed framework. We lay out the motivation that inspired our framework design, and present the notations that we use in the remainder of this thesis. Section 3.3 details each task we tackle and our proposed methods to tackle such tasks. Section 3.4 lays out possible use cases of our framework, depending on the type of information provided to the framework and the end-users' expectations. Finally, in Section 3.5, we provide a summary of our framework proposed in this chapter.

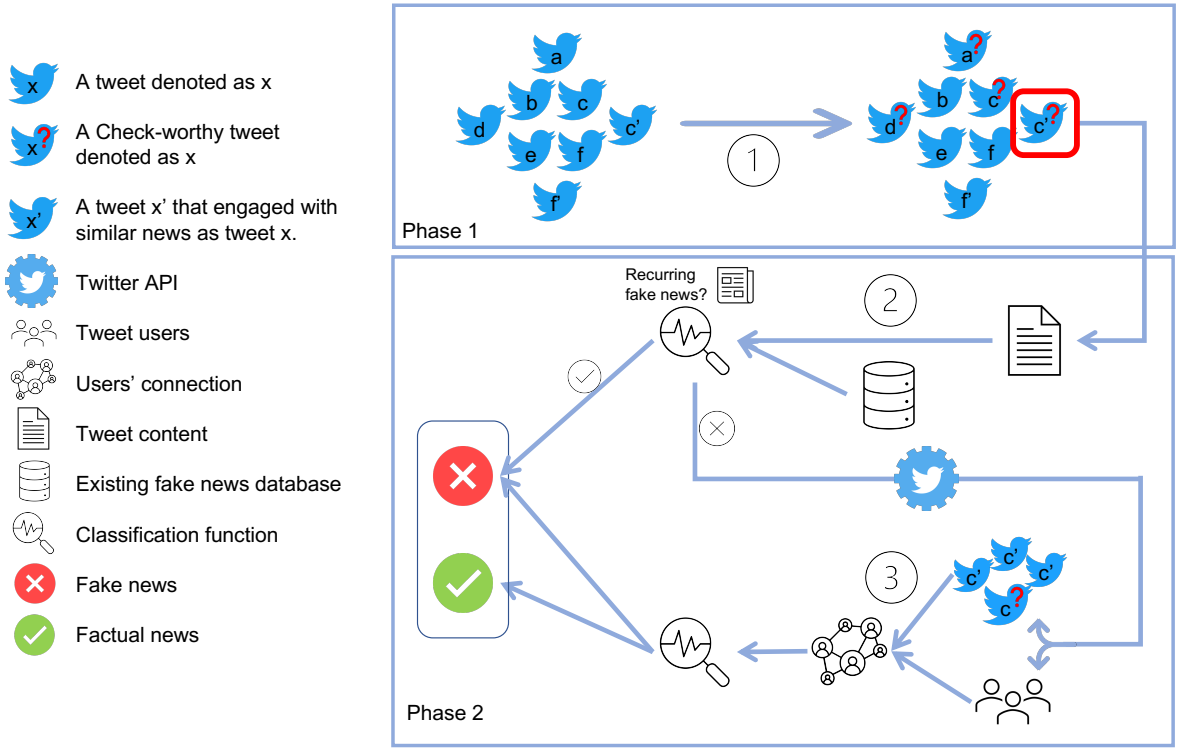


Figure 3.1: Our proposed Fake News Detection Framework. Phase 1 is the *Check-Worthiness Detection Phase*. Phase 2 is the *Fact-Checking Phase*. Task 1 in Phase 1 aims to *rank tweets and sentences based on their check-worthiness*, Task 2 in Phase 2 is dedicated to *identify recurring fake news*, Task 3 in Phase 2 focuses on *using Twitter network in identifying fake news*. Finally, the framework return the predicted non-factual information within tweets and sentences to the end users, i.e., the fact-checking journalists, general public.

3.2 Motivation and Preliminaries

It is infeasible to fact-check every tweet that is being published, as around 9530 tweets are being published[1] every second, while some of these tweets also focus on news articles being published, and fact-checking all tweets requires a large amount of computational

[1] https://www.internetlivestats.com/twitter-statistics/

power. Moreover, some fake news contains recurring themes and topics, that have been debunked previously [55], while newly emerged fake news can easily spread through conspiracy theorists[19]. Inspired by these obstacles, we propose a fake news detection framework (FNDF). Figure 3.1 illustrates our proposed framework, FNDF, which consists of two phases and three tasks in total. Specifically, Phase 1 of FNDF contains only Task 1. Task 1 aims to prioritise tweets and/or claims that require fact-checking, from a large number of published tweets/claims. Thus, Task 1 allows us to focus more on the tweets and news/sentences that are potentially spreading false information. Phase 2 of FNDF contains two tasks, Task 2 and Task 3. Task 2 aims to fact-check the potentially misleading claims in tweets and sentences by comparing these sentences and tweets with existing fake news, while Task 3 employs the Twitter network to assist the identification of fake news. Thus, our framework first aims to reduce the number of claims that needed to be fact-checked, by identifying suspicious and check-worthy claims, before going through the computationally expensive analysis process. Then, our framework aims to use textual information from the claims, as well as use features that are readily available on Twitter to identify fake news. As such, our proposed FNDF does not require the spreading pattern (how the fake news is spreading on Twitter, such as retweets, likes, and replies) of news to detect fake news, thus aiding the early detection of fake news before any fake news is widely spread. Thus, our framework can enable large-scale scanning of information published online to identify the claims that require fact-checking. Our framework also enables automatic recurring fake news labelling while providing debunking information based on previously debunked fake news, as well as allowing early detection of fake news among small conspiracy groups when the misleading information has not spread widely. Table 3.1 presents the notations we use in this thesis.

3.2.1 Phase One (P1) - Worth-Checking Ranking (WCR) Phase

The number of tweets generated per day would require heavy labour to manually screen for fact-checking, and similarly, would be difficult for automated fact-checking systems to cope with. Moreover, Narrowing down a list of claims to further fact-check is a common practice adopted by fact-checkers [59, 160]. Thus, in order to reduce the number of tweets that requires fact-checking and to address **Gap 2** discussed in Chapter 2, this phase aims to develop a model to help identify the most check-worthy claims from tweets and news articles based on the possibility of them being related to false statement. This phase contains one task (Task 1), which assesses and ranks claims from tweets and articles based on their check-worthiness.

For example, Figure 3.2 shows that among 70 debates and speeches given by the US presidential candidates between 2012 and 2017, more than 95% of them have less than 0.1% of check-worthy sentences, while more than half have no more than 0.02% of check-worthy

Table 3.1: Notations used in this thesis.

Notation	Definition
X	A set of sentences and tweets
x	A sentence or tweet
$X_{checkworthy}$	The set of check-worthy claims identified from X
X_{fake}	The set of claims and tweets identified as fake
T_x	The set of tweets related to x
T	The set of tweets of all T_x for $\forall x \in X$
t	A tweet in the set of tweets T_x
e	An entity
E	A set of entities
e_{pair}	A pair of two entities
$\overrightarrow{e_{pair}}$	The vector representation of a pair of entities extracted from x
FN	An existing fake news collection
fn	An identified fake news in the set of identified fake news FN
U	The set of users who posted the set of tweets T
U_x	The users that engaged with x, who posted the tweets T_x
u	A user in the set of users U or U_x who posted t
$\overrightarrow{u}$	The modelled vector representation of user u
G	The graph that consists of users U and their friends or followers on Twitter

sentences, according to the journalists [7, 11]. We do not report the proportion of check-worthy tweets on the Twitter platform, because there are no reasonable data to perform such analysis.

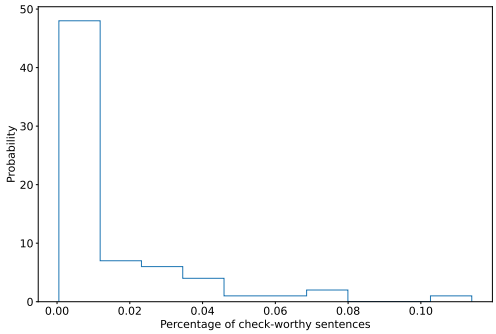


Figure 3.2: In the CLEF CheckThat! 2019 task 1 dataset [7], among some selected 70 debates and speeches transcripts given by the US presidential candidates between 2012 and 2017, 95% of them have less than 0.1% check-worthy sentences.

The process of Phase 1 can be describe as follows:

$$X_{checkworthy} = f_{checkworthy}(X) \tag{3.1}$$

The aim of this phase is to identify the best function $f_{checkworthy}()$ that can be used to score

claims from tweets and sentences X by their check-worthiness, and identify a set of check-worthy tweets/sentences $X_{checkworthy}$.

3.2.2 Phase Two (P2) - Fact-Checking (FC) Phase

Each identified check-worthy claim needs to be classified as containing non-factual information or not. Recall Sections 2.4.2 and 2.4.3, where we introduced two types of existing methods in detecting fake news, i.e., identifying recurring fake news, and leveraging Twitter information in identifying fake news. Similarly, this phase aims to tackle the classification of check-worthy claims as containing non-factual information or not in two tasks: respectively, Task 2 identifies recurring fake news and Task 3 uses Twitter's user network structure to identify fake news. Specifically, the aim of P2 is to classify a claim as fake news or not, using the textual features and user network features extracted from the claim and its twitter engagement. The aim of P2 can described as follows:

$$\widehat{Y_x} = factcheck(x, T_x, U_x) \tag{3.2}$$

where $\widehat{Y_x}$ is the conclusion if the identified check-worthy sentence/tweet x contains a factual claim or not, while $factcheck()$ is the classification function that fact-checks a tweet or sentence x, based on the associated tweets/sentences T_x and their engaged users U_x.

3.3 Individual Tasks and Proposed Methods

In this section, we formally define the three tasks we aim to tackle in each step of our proposed FNDF, as well as introduce the main methods we use in tackling the three tasks.

3.3.1 Task One (T1): Assessing and Ranking Check-Worthiness of Sentences and Tweets

In order to focus available computing power on the most check-worthy claims, among all the tweets and sentences fed into our framework, we aim to assess the check-worthiness of all the tweets and sentences, and rank them in descending order based on their check-worthiness. Thus, this task aims to address **Gap 2** discussed in Chapter 2, which states that there is limited research on whether language models can be combined with entity information in identifying check-worthy claims. This task is the only task in P1, thus the task definition is the same as the P1 definition presented in Equation (3.1).

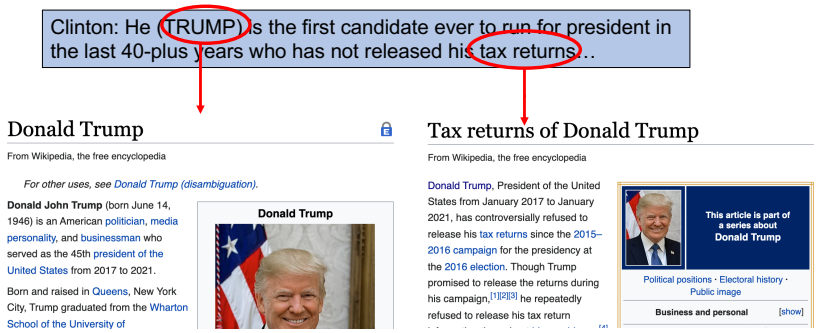


Figure 3.3: An example showing two entities in a check-worthy sentence that are related to each other.

To tackle this task, we propose to use deep learning methods and information from entities presented in a sentence/tweet x. We obtain information about entities using an embedded knowledge base, and compute the relationships between a pair of entities present in the text. We then combined the entity pair representation with the textual representation of the given sentence/tweet x to predict whether x is check-worthy or not. For example, Figure 3.3 illustrates a claim that can be identified as check-worthy by combining entities information with language model representations, where the entity *Donald Trump* and *Trump's tax return* are closely related to each other in the Wikipedia knowledge base, and can give important information within the context of this claim.

To analyse entity information together with the textual information, we first extract a set of entities E appearing in a given tweet/claim x:

$$E = Extract(x) \tag{3.3}$$

Each entity $e \in E$ then is presented in a lower-dimensional space as embeddings by function ent_emb():

$$\overrightarrow{e} = \text{ent\_emb}(e) \tag{3.4}$$

A pair of entities e_{pair} is then represented as a vector using a combination method combine():

$$\overrightarrow{e_{pair}} = \text{combine}(\overrightarrow{e_1}, \overrightarrow{e_2}) \tag{3.5}$$

We then compute the check-worthy score of the tweet or claim x using x and the a representation for a pair of entity extracted from x, as follows:

$$\widehat{x_{checkworthy}} = cw(x, \overrightarrow{e_{pair}})) \tag{3.6}$$

where the $cw()$ function decides the check-worthiness ($\widehat{x_{checkworthy}}$) of the input x, and $\overrightarrow{e_{pair}}$

is the entity pair representation for a pair of entities in x.

3.3.2 Task Two (T2): Assisting Fake News Detection using Previous Debunked fake News

As suggested by [60], some rumours and fake news may reappear after being debunked. Thus, this task aims to identify non-factual claims and tweets that have resurfaced after being debunked (cf. **Gap 3** discussed in Chapter 2). The task is formulated as a classification task:

$$\widehat{Recurring}FN_x = cls(x, FN) \tag{3.7}$$

where $\widehat{Recurring}FN_x$ is the model output of the classification function $cls()$ for a given tweet/claim x requiring fact-checking. To tackle Task 2, we propose to identify recurring fake news by calculating the entailment of x with every fake news fn in an existing fake news collection FN. If x entails any known fake news (that is not time-sensitive), we conclude that x is a recurring fake news, otherwise not. Specifically, for each x, the entailment relation of x and fake news $fn \in FN$ can be classified as *agree*, *unrelated*, or *disagree*[2]:

$$entailment(x, fn) \rightarrow \{agree, unrelated, disagree\} \tag{3.8}$$

The function f_{entail} aims to classify the entailment relationship between x and fn:

$$entailment(x, fn) = f_{entail}(x, fn) \tag{3.9}$$

$$\tag{3.10}$$

We calculate the entailment relationship between a given x with all $fn \in FN$:

$$RecurringFN_x = \begin{cases} 1, \text{if } \exists\ fn \in FN,\ f_{entail}\ (x, fn) = agree \\ 0, otherwise \end{cases} \tag{3.11}$$

where if a tweet/claim x entails any fn, we conclude that x is indeed non-factual and recurring ($RecurringFN_x = 1$). On the other hand, if x does not entail any fn, we conclude that x is not a recurring non-factual fake news ($RecurringFN_x = 0$), and will continue to Task 3.

[2]An example of a claim x agreeing a debunked fake news is as follows: statement A *"vaccine causes autism"* agrees with statement B *"according to a study published in the Lancet, getting MMR vaccine correlates to a higher rate of autism observed in children"*, thus we consider statement A entails statement B.

3.3.3 Task Three (T3): Social Network Structure Assisted Fake News Detection

Fake news can rapidly spread on social media [87] because users are exposed to their connecting users' activities. It may be reasonable to assume that the wisdom of the crowd may provide us with some information regarding whether a claim is fake. As discussed in Section 2.1, the echo chamber effect has been observed on social media platforms, which inspired us to hypothesise that some non-factual tweets may have originated and spread within some specific groups that have interests in the topics and theories that the tweets convey. Moreover, we identified **Gap 4** in Section 2.4.4, which stated that we need to identify the most effective way to use the dynamic social media users' connections with each other in detecting fake news. Thus, in Task 3, we focus on identifying newly emerged fake news that begins circulating on social media. Specifically. we aim to study the user network structure on social media, and use such user network structure to assist the identification of non-factual claims and tweets from previously identified check-worthy claims and tweets$X_{checkworthy}$. Specifically, Task 3 aims to classify if a check-worthy claim x contains non-factual information, using a set of engaging users U_x that tweeted the engaging tweets T_x, so as to address **Gap 4**. Thus, Task 3 is defined as follows:

$$\widehat{Y_x} = cls(x, T_x, U_x) \tag{3.12}$$

That is, for each check-worthy claim x not containing resurfacing misinformation, we classify it as being factual or not, using its engaging tweets T_x and engaging Twitter users U_x. We instantiate the tweet retrieval step in our FNDF using existing methods provided by the Twitter API[3]. We propose to use the social network structures of the users to analyse the set of tweets T_x related or engaged with x. In particular, we first propose to generate user embeddings from the social network structure. We then use the users' embeddings of the users engaged with the tweet/claim x, to classify if the tweet/claim x is indeed fake news.

Specifically, for each x identified as check-worthy, we analyse the users U_x that engaged with the tweets T_x, using the users' network (G) to obtain the engaging users social network embedding $\overrightarrow{U_x}$:

$$\overrightarrow{U_x} = \underset{u \in U_x}{\text{aggregate}}(\text{UserRepresentation}(u, G)) \tag{3.13}$$

where the UserRepresentation() is the model that represents a user u as vector based on the user network G, and aggregate() is the aggregation model that aggregate all the user embeddings to obtain a single vector for all the engaging users U_x.

[3] https://developer.twitter.com/en/docs

The final classification result $\widehat{Y_x}$ is then derived from the classification function $SN_{cls}()$, with the social embedding $\overrightarrow{U_x}$ of the users U_x engaged with x as input:

$$\widehat{Y_x} = SN_{cls}(\overrightarrow{U_x}) \tag{3.14}$$

3.3.4 End-to-End Evaluation

Finally, we evaluate the end-to-end performance of our proposed FNDF, which classifies if a tweet/sentence contains fake claims or not, and return the fact-checked results back to users. Figure 3.1 illustrates our end-to-end user model. The end-to-end use case of FNDF is to identify the set of fake information X_{fake}, given a set of tweets and sentences X, using functions (3.1) and (3.2) in turn. As such, we evaluate the effectiveness of FNDF in correctly identify the set of X_{fake} from the set of tweets and sentences X that enters our framework.

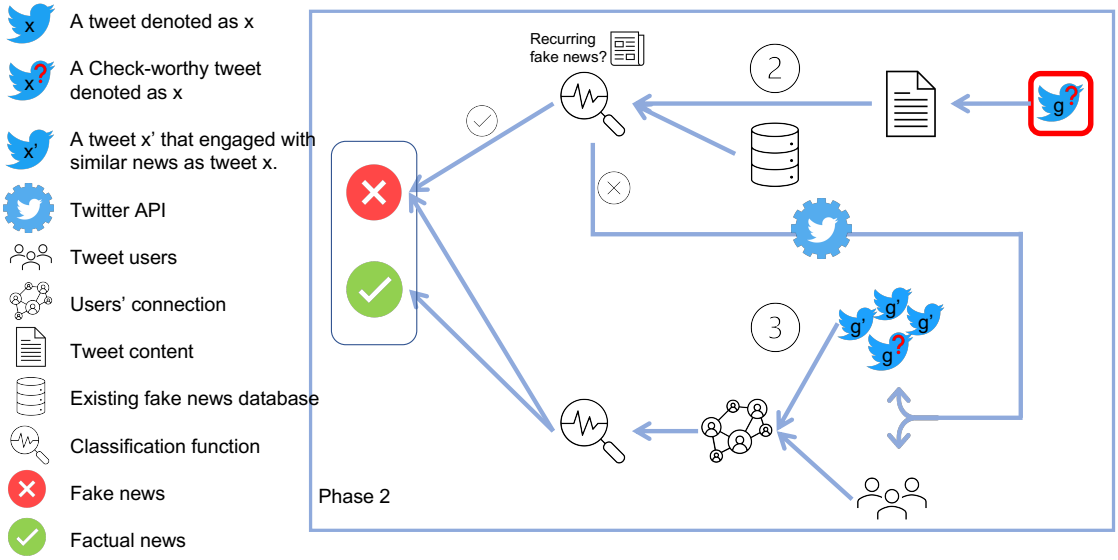


Figure 3.4: Framework structure for the individual tweet fact-checking scenario. This instance of FNDF uses Task 2 (comparing the current claim with fake news database) and Task 3 (Twitter network analysis) to decide if the given tweet contains fake news.

3.4 Possible Use Cases

As mentioned in Section 3.2, our framework can be used in multiple scenarios, where not every component is necessary. Our framework can be used to fact-check individual tweets, or to screen tweets posted daily and identify fake news, depending on the expectations of an end-user. Thus, we provide the possible use cases one tweet may go through based on the user requirements as follows:

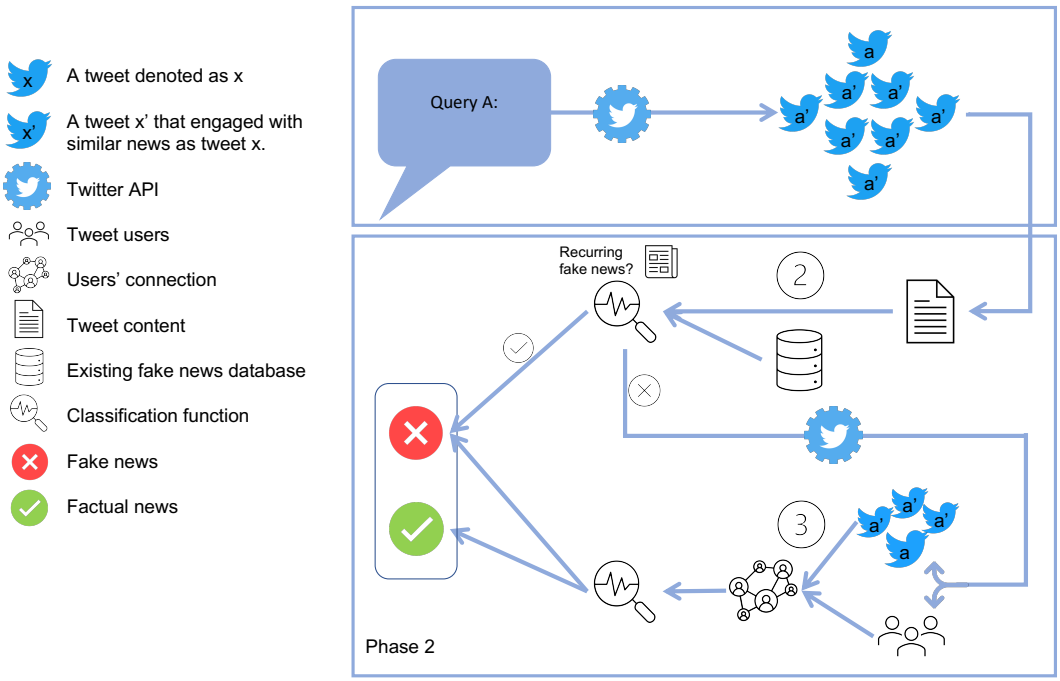


Figure 3.5: Framework structure for the query-related fact-checking scenario. This FNDF instance uses the Twitter API to retrieve a range of tweets that are related to the query, using Task 2 (comparing the current claim with fake news database) and Task 3 (Twitter network analysis) to identify fake news contained in the retrieved tweets

- **Scenario 1: Batch of tweets fact-checking.** The batch of tweets fact-checking scenario is where a user would like to monitor all tweets being created and shared online, and identify any tweets that contain non-factual information. This is a Twitter monitoring use case, which requires the full framework to work together, thus Figure 3.1 represents the framework used for this scenario.

- **Scenario 2: Individual tweet fact-checking.** Our platform can also fact-check an individual tweet. For example, if a user has already identified a tweet they want to fact-check, they can use our proposed platform to identify if this given tweet contains fake news or not. When given a single tweet, our platform can omit Phase 1, the Worth-Checking Ranking Phase - since the user already decided that such tweet requires fact-checking, and only focus on Phase 2, the Fact-Checking Phase of our FNDF. Figure 3.4 presents the components of our framework that are used for this scenario.

- **Scenario 3: Query-based tweets fact-checking.** Using the standard Twitter search function, our framework can also identify fake news within a certain topic. The search feature provided by the Twitter API allows a user to search for a set of tweets using a query, and our framework is able to work within this limited set of tweets. Specifically, our framework is able to retrieve a set of tweets using the given query, and identify if the returned tweets contain any non-factual claims. Figure 3.5 illustrates the

framework structure that aims to handle such scenario.

3.5 Conclusions

In this chapter, we have introduced our proposed framework, FNDF, which uses a wide range of features of a tweet/claim to identify if a tweet/claim contains false information. We introduced the motivation to include two phases and three tasks we aim to focus on in our framework. We also formally defined these tasks, and presented the possible use cases where our framework can be used in real-world scenarios. In particular:

- Section 3.2 introduced the motivation and preliminaries of each phase. We showed that Phase 1 of our FNDF aims to assess a large amount of tweets and claims by their check-worthiness; if a tweet/claim is check-worthy, it will enter Phase 2 of our FNDF, which fact-checks if the check-worthy claim contains false information.
- Section 3.3 introduced three tasks in our FNDF. We presented the aim for each task, defined each task in our framework, and described the high-level description of how we aim to tackle each task. Specifically, Task 1 aims to address **Gap 2** presented in Section 2.4.4 by combining entity representations with language representations to more effectively assess the check-worthiness of a given sentence/tweet. Task 2 aims to address **Gap 3** by identifying recurring fake news using existing fake news collections. And Task 3 aims to address **Gap 4** by identifying fake news using user network structure on Twitter.
- Section 3.4 introduced three possible use cases of our framework, FNDF, and demonstrated how our proposed framework is able to adapt to each one of them.

In the remainder of this thesis, Chapter 4 addresses Task 1, which aims to identify the most check-worthy tweets and claims; Chapter 5 addresses Task 2, which aims to identify recurring fake news using existing fake news collections; and Chapter 6 addresses Task 3, which aims to use Twitter's network structure to identify tweets and claims that contain non-factual information.

In the next chapter, we describe the experiments that aim to tackle Task 1 – assessing a sentence/tweet's check-worthiness, and addressing **Gap 2**.

Chapter 4

Assessing and Ranking Check-Worthiness of Claims

4.1 Introduction

In Section 2.4, we described the task of identifying the most *check-worthy* sentences and tweets, and surveyed the recent works that focused on tackling this task. For example, the ClaimBuster system [71] was trained to label sentences in a news article as "non-factual", "unimportant factual", or "check-worthy factual". The recent CLEF' 2019, 2020 & 2021 CheckThat! Labs [6, 34, 124] were introduced as shared evaluation forums where participants were tasked to rank texts based on their estimated check-worthiness. Section 2.4.1 described how it is common to apply neural language models to represent sentences and tweets [33, 41, 52, 69, 78, 113, 114, 124, 156].

As described in Section 1.1, to make a *claim* is to assert that something is true[1]. A claim usually contains a subject and/or an object [9], where the subject and the object are often entities [155]. Thus, the entities presented in claims are vital in analysing the claim, and can determine if the claim is check-worthy or not. Moreover, claims made by politicians during debates and by users posted on Twitter often contain information about established entities (for instance, entities that are documented in Wikipedia) [8]. For example, Figure 3.3 in Section 3.3.1 shows an example where the two entities highlight the important components of the sentence, and are related to each other, which can help to identify the sentence as check-worthy. Thus, in this chapter, we focus on analysing established entities in a claim in our check-worthiness identification task, as established entities can be verified with documented information, such as knowledge graphs.

[1] In everyday usage, a sentence or a tweet usually asserts only one statement, inspiring us to consider a sentence and a tweet as valid forms of claims.

As mentioned in Section 2.7.1, knowledge graphs (KGs) are useful sources of information about entities, particularly how they relate to each other. Typically, entities and their relationships are represented using a triplet structure ($\langle entity_h, r, entity_t \rangle$) in a knowledge graph. An example of such a triplet is $\langle$*Arizona*, *a_state_of*, *the_United_States*$\rangle$. Recent works [16, 180, 194] have shown that learned embeddings can be derived from KGs, allowing for the advantages of word embeddings to be applied to the entities found in the KGs.

Recall the thesis statement introduced in Section 1.3, where we introduced the first phase of our framework being the identification of check-worthy claims from tweets and sentences, and hypothesised that analysing **embedded entities** in texts can help more accurately **identify check-worthy claims**, from tweet content, articles, and debate quotes. This chapter aims to test this hypothesis, by tackling Task 1 (defined in Section 3.3.1), the task that aims to assess and rank the check-worthiness of sentences/claims and tweets. Specifically, we propose to combine language models with entity embeddings for enhancing the performance of identifying check-worthy sentences and tweets.

This chapter aims to address **Gap 1** and **Gap 2** identified in Section 2.4.4. Specifically, **Gap 1** states that for the task of identifying check-worthy sentences and tweets, there are no research that have identified the most suitable language model, and **Gap 2** states the need for research on how to combine entity information with sophisticated language representations. We address these two gaps by addressing the detailed **Limitations L1 & 2** identified in Section 2.6.4 and **Limitations G1-3** identified in Section 2.7.1.4.

In particular, **Limitation L1** recognises the need to identify the best language model for each task in this thesis. We address **Limitation L1** by conducting experiments using six widely used text analysis models, to identify the most suitable language model for the identification of check-worthy sentences and tweets task.

Limitation L2 identifies the need to enrich language models with additional entity information, while **Limitation G1** identifies the need to test if embedded entities are beneficial, in the task of detecting fake news. To address these two limitations, we hypothesise that the embedded entity vectors obtained from KG embeddings (*entity embeddings*) can improve the identification and ranking of check-worthy sentences and tweets, and run experiments to verify such a hypothesis. Thus, we propose a novel model to represent a sentence or a tweet, by combining a neural language model with an entity pair representation for each pair of entities in the sentence or tweet. We conduct experiments to test the hypothesis, and show that enriching language representations with entity representations are indeed beneficial in detecting check-worthy sentences and tweets.

Limitation G2 states that the current entity embeddings trained from knowledge graphs are not tailored to be used to represent entity pairs, and are not fine-tuned on the check-worthy tweets and sentences identification task. To address **Limitation G2**, we design and study two

types of methods to represent a pair of embedded entities together, which calculate the pairwise vector product and concatenate two entity embedding together. We show that using entity pairs to represent a tweet/sentence allows us to capture rich information from both entities present, and the potential relationships of these two entities.

Finally, **Limitation G3** states that it is unclear which KG entity embedding method is the most effective at representing entities within sentences and tweets, to most accurately identify the check-worthy sentences and tweets. Thus, to address **Limitation G3**, we propose to compare the performances of six different types of KG entity embedding models (representing six types of well-used graph embedding methods). We show that the ComplEx model [180] can produce the most accurate results in identifying check-worthy news from sentences and tweets through extensive experiments.

The rest of the chapter is structured as follows: Section 4.2 states the task problem, along with our proposed model to address the task. We present our experimental setup in Section 4.3, and show the results of the experiments in Section 4.4. Finally, we provide concluding remarks in Section 4.5.

4.2 Check-Worthiness Prediction using Entity-Assisted Language Models (Phase 1 Task 1)

In this section, we expand on the definition of Task 1 presented in Section 3.2.1, to tackle and introduce our proposed entity-assisted language model in detail.

4.2.1 Check-Worthiness Prediction Task

We aim to tackle the task of identifying the set of check-worthy tweets/sentences from a given set of tweets/sentences. Table 4.1 presents the notations (a subset of the notations defined in Table 3.1, and notations specific for this chapter) used in this chapter.

Task 1 is stated in Equation (3.1) as:

$$X_{checkworthy} = f_{checkworthy}(X) \tag{4.1}$$

where $f_{checkworthy}()$ is the function that identifies check-worthy claims $X_{checkworthy}$ from X. Note that this task can be formulated as a classification task, aiming to predict (denoted $\hat{y_i}$), for each sentence/tweet, whether a human would label it as check-worthy or not (c.f. y_i). In this classification task $f_{checkworthy}()$ is the classification function that classifies whether a given sentence/tweet x is check-worthy. The task can also be formulated as a ranking task,

Table 4.1: Notations used in Chapter 4.

Notation	Definition
X	A set of sentences and tweets that enter our framework
x	A sentence or tweet in the set of sentences and tweets X
$X_{checkworthy}$	The set of check-worthy claims identified from X
e	An entity
e_h	The head entity in a triplet $\langle entity_h, r, entity_t \rangle$
e_t	The tail entity in a triplet $\langle entity_h, r, entity_t \rangle$
$\overrightarrow{x}$	The vector representation of x
$\overrightarrow{e_h}$	The vector representation of e_h
$\overrightarrow{e_t}$	The vector representation of e_t
$\overrightarrow{e_{pair}}$	The vector representation of the entity pair $< e_h, e_t >$

such that the predicted most check-worthy $x \in X$ are ranked highest – indeed, this is the task formulation taken by the CLEF' 2019 and 2020 CheckThat! Labs [6, 11]. In this ranking task, $f_{checkworthy}()$ is a ranking function that ranks X based on the check-worthiness of each x. In our present study, we propose a uniform model, which addresses the estimation of check-worthiness both as classification and ranking tasks, when measuring the effectiveness of our models.

Our proposed uniform model for tackling the identification of the check-worthiness of each sentence/tweet consists of two components: text representation through the use of language models, and an entity pair[2] representation obtained from entity embeddings – discussed further in Section 4.2.2. Each sentence/tweet is represented by a single embedding obtained from a language model (denoted by $LM()$), which is discussed further in Section 4.2.3. There are three steps involved in representing a pair of entities appearing in the text:

1. Resolving all entities that appear in the text to the corresponding entity using entity linking [36, 116];

2. Transforming the resolved entities into dense entity embeddings through the application of KG embeddings (denoted by $KG()$) – we discuss the choice of KG embeddings in Section 4.2.4;

3. Each pair of entity embeddings are combined through a combination method (denoted by $COM()$) to form a single representation for the entity pair. Note that, for a sentence/tweet that contains more than two entities, every two entities form an entity pair.

[2]We also experimented with text representation combined with a single entity, and sentence/tweet combined with three entities, and neither perform well in this task. For ease of reading, we do not present the equations and experiments for such structures.

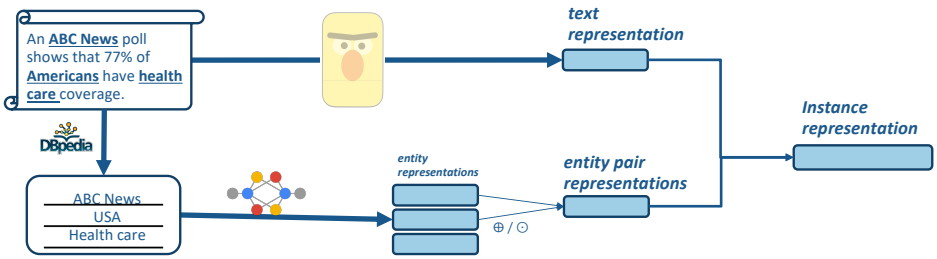


Figure 4.1: Our proposed Entity-Assisted Language Model.

4.2.2 Overall Structure of Our Proposed Model

In order to leverage the semantic representation of various language models, as well as the entities in text, for each sentence/tweet, we propose to combine its language representation along with an entity pair representation for each pair of entities in the text using a language model.

Firstly, for a sentence/tweet x in which a set of entities $E(x)$ have been identified through the application of an entity linker, our model forms pairs of entities with every two unique entities, thus we consider *input instances* ins_i, based on pairs of distinct entities:

$$ins_i \in \{\langle x, e_h, e_t\rangle \, \forall \, \langle e_h, e_t\rangle \in E(x) \times E(x)\} \tag{4.2}$$

where e_h and e_t are the head and tail entities. For ease of notation, let $ins_i \in x$ denote a particular instance ins_i obtained from x using Equation (4.2). Then, given an input instance ins_i, we develop two separate models for the $f_{checkworthy}()$ presented in Equation (3.1): $f^{cls}(ins_i)$ for text classification and $f^{rank}(ins_i)$ for ranking. Furthermore, for combining the embeddings of a given entity pair, we use two different methods as explained below.

In particular, Figure 4.1 shows the architecture of our proposed model. In the input stage, we use the text as input to a language model $LM()$, so as to obtain the text representation of the input text $\overrightarrow{x}$, i.e.:,

$$\overrightarrow{x} = LM(x) \tag{4.3}$$

For **each entity pair** e_h and e_t in an input instance, we represent the entity pair as $\overrightarrow{e_{pair}}$ in a high dimensional space. Thus, we firstly use an existing KG embedding model $KG()$ to extract entity embeddings $\overrightarrow{e_h}$ and $\overrightarrow{e_t}$ for entities e_h and e_t. Next, we use a combination method $COM()$ to obtain the **entity pair** representation $\overrightarrow{e_{pair}}$. Specifically, as combination methods we use the **vector element-wise product operation** (denoted by emb_prod()), or the **vector concatenation operation** (denoted by emb_concat()), or the entity similarity and relatedness score proposed by Zhu and Iglesias [209] for each entity pair[3] (denoted by similarity()). This

[3]For brevity reasons we do not describe these methods in detail, as it is described by Zhu and Iglesias [209].

process can be represented as follows:

$$\overrightarrow{e_h} = KG(e_h), \overrightarrow{e_t} = KG(e_t) \tag{4.4}$$

$$COM() \in \{\text{emb\_prod}(), \text{emb\_concat}(), \text{similarity}()\} \tag{4.5}$$

$$\overrightarrow{e_{pair}} = COM(\overrightarrow{e_h}, \overrightarrow{e_t}) \tag{4.6}$$

It is of note that we select emb_prod() and emb_concat() because of their wide use as neural operators for combining two vectors (e.g., [29, 40]). To address G2, we compare emb_prod() and emb_concat()'s performance in combining entities and representing entities in a sentence. As emb_prod() and emb_concat() aim to tailor the entity pair representation to better suit the task of identifying check-worthy tweets and sentences.

We combine the text representation and entities pair, to form the input instance representation $\overrightarrow{ins_i}$, by concatenating the language representation $\overrightarrow{x}$ with the entity pair representation $\overrightarrow{e_{pair}}$[4]:

$$\overrightarrow{ins_i} = \overrightarrow{x} \oplus \overrightarrow{e_{pair}} \tag{4.7}$$

Next, $\overrightarrow{ins_i}$ can be used both as part of a classification $f_{cls}()$ or a ranking $f_{rank}()$ task. In our experiments, $f_{cls}()$ is a fully connected layer with a softmax activation function that estimates the class likelihood $\widehat{y^{cls}_{ins_i}}$, while $f_{rank}()$ is a fully connected layer with a sigmoid activation function to obtain the check-worthiness score $\widehat{y^{rank}_{ins_i}} \in (0, 1)$ for ranking the texts in descending order:

$$\widehat{y^{cls}_{ins_i}} = f^{cls}(\overrightarrow{ins_i}) = \frac{e^{((\overrightarrow{ins_i} \otimes k)+b)}}{\sum_j e^{((\overrightarrow{ins_i} \otimes k)+b)}} \tag{4.8}$$

$$\widehat{y^{rank}_{ins_i}} = f^{rank}(\overrightarrow{ins_i}) = \frac{1}{1 + e^{((-\overrightarrow{ins_i} \otimes k)+b)}} \tag{4.9}$$

where k denotes a fully connected layer kernel and b denotes bias. The objective of our experiments is to identify the most effective $f^{cls}()$ and $f^{rank}()$ models, for classifying and ranking check-worthy texts, respectively.

A sentence/tweet may contain more than one pair of entities, with corresponding different levels of check-worthiness. For these cases, we assume that as long as at least one pair of entities is check-worthy within a sentence/tweet, the sentence/tweet is check-worthy. Thus, the obtained $f^{cls}()$ and $f^{rank}()$ models are applied for each pair of entities in the text. Hence, to obtain the final check-worthiness of a given text, we take the maximum check-worthiness

[4]We use a uniform $[-1, \ldots, -1]$ vector to represent any entity not having any embedding in the pre-trained KG embeddings.

label/score across all pairs as follows:

$$\widehat{y_x^{cls}} = \max(f^{cls}(\overrightarrow{ins_i})) \; \forall \overrightarrow{ins_i} \in x \tag{4.10}$$

$$\widehat{y_x^{rank}} = \max(f^{rank}(\overrightarrow{ins_i})) \; \forall \overrightarrow{ins_i} \in x \tag{4.11}$$

where $ins_i \in x$ denotes an input instance ins_i occurring in text x, and $\max()$ denotes that we take the highest score/likelihood among all $ins_i \in x$ as the final check-worthy score/likelihood for x.

4.2.3 Language Models

As presented in Section 2.6, there are many ways to represent text, such as BoW models and neural network models. In order to study the effectiveness of using language models in identifying check-worthy tweets and sentences, and addressing **Limitation N1**, we evaluate 3 groups of language models. First, TF.IDF vectors are used as a representative of traditional BoW models. Second, we use a BiLSTM model with an attention mechanism to represent the non-pre-trained language models. Finally, we use several BERT-related neural language models (BERT, ALBERT, RoBERTa, and BERTweet) to represent the current state-of-the-art pre-trained language models. We combine these language model representations with the entity pair representations, to study the robustness of using entity embeddings across different types of language models.

4.2.4 Obtaining Entity Embeddings and Similarity from KG Embedding Models

Section 2.7.1 introduced multiple ways to analyse entities appearing in the text. Previous studies have shown that some entity embedding methods can benefit the identification of check-worthy material. For example, Gąsior et al. [54] used named entity recognition to identify the types of entities present in a sentence (e.g. person, location, organisation, money) as hand-crafted features, and showed that it can improve the performance of TF.IDF. Ciampaglia et al. [30] showed that the *graph distance* between two entities within a KG (i.e. the number of steps on the graph to reach one entity from another) could be used to improve fake news detection accuracy when applying an entity linking method on news articles. However, using graph distance considers only the number of hops between two entities within the knowledge graph, and therefore does not address other possible relationships between the entities (e.g. a person (an entity) being *the president* of a country (another entity)). This means that using only the KG's ontology structure results in less information compared to using embeddings that may capture more entity relationships.

Table 4.2: Examples of the most similar entities to Barack Obama, using each of the KG embedding models.

Embedding Model	Most similar entities to *Barack Obama*, in descending order from left to right		
Wikipedia2Vec	Michelle Obama	John McCain	US presidential election
TransE	Women's History Month	A Child's History of ...	Thickness network ...
TransR	Executive Order 13654	BODY SIZES OF ...	ynisca kigomensis
RESCAL	Neural representation ...	Natalie Grinczer	Octavia E. Butler
DISTMult	live preview	Neonatal peripherally ...	KSC - STS-3 Rollout ...
ComplEx	Peter B. Olney	James Willard Hurst	Robert H. McKercher

Thus, to address **Limitations L2 and G1**, we instead propose to obtain the entity representations using a range of KG embedding models (i.e., Wikipedia2Vec [194], TransE [16], TransR [180], RESCAL [100], DISTMult [129], and ComplEx [195]). We believe that using KG embedding models allows us to acquire the implicit and hidden KG-based relationships between two entities that are encoded in the embedding vectors that have been learned by a particular model. Moreover, following Ciampaglia et al. [30][5], we focus on using pairs of entities in analyse the sentence – entities informations.

Different KG embedding models can return varying results when given the same entity and task. For example, Table 4.2 shows the most similar three entities for the United States' President $\langle$*Barack Obama*$\rangle$ obtained using six different KG embedding models that we use in this study. Specifically, Wikipedia2Vec returns the entities that appear closer to the entity *Barack Obama* in the sentence, while the other 5 models show a variety of very specific entities that *Barack Obama* has a relationship with (e.g., the law he passed, the article he wrote, the person he attended the same school with). Such differences in the output provided by the KG embedding models are due to the varying datasets used to train the models. Moreover, the different KG embedding models can result in different performance in identifying check-worthy sentences and tweets. Thus, to address **Limitation G3**, we compare the above mentioned six KG embedding models, to identify the most effective KG embedding model in identifying check-worthy tweets and sentences.

Therefore, the key argument of this chapter is that by including the entity embeddings $\overrightarrow{e}$ for each entity e (appearing in the sentence) into our models, we are able to consider the KG-based network relationships of entities in a sentence, when making predictions about the check-worthiness of a sentence. Indeed, entities that are far apart on a simpler word embeddings space may be closer on the entity embedding space, and combining the word embeddings and entity embeddings may be able to bring these two types of information together. Overall, this provides more evidence about the expected co-occurrence of different types of entities within a sentence for identifying those sentences requiring fact-checking.

As a baseline comparison, we also calculate the similarity of entities following the method

[5]Indeed, as we later show in Section 4.3, texts containing 2-4 entites are the most frequent in this dataset.

described by Zhu and Iglesias [209]. Thus, we obtain two scores for each entity pair – similarity score and relatedness score – to represent an entity pair.

4.3 Experimental Setup

Our experiments address four research questions that concern both the check-worthy sentences detection and check-worthy tweets detection, as follows:

- **RQ 4.1:** Do BERT-related language models outperform the TF.IDF and BiLSTM baselines in identifying check-worthy sentences/tweets? This research question aims to address the **Limitation L1** presented in Section 2.6.4, which concerns finding the best language models for the identification of check-worthy claims.
- **RQ 4.2:** Does the use of **entity embeddings** improve the language models' F1 score in identifying check-worthy sentences/tweets? This research question aims to address **Limitations L2 & G1** presentend in Section 2.6.4 and Section 2.7.1.4 respectively, which concern enriching language models with additional embedded entity information.
- **RQ 4.3:** Which combination method, $COM() \in \{\text{emb\_prod}, \text{emb\_concat}\}$, performs the best in improving the performance of text representations at identifying check-worthy sentences/tweets? This research question aims to address **Limitation G2**, which concerns tailoring entity embeddings to better assist language models.
- **RQ 4.4:** Among Wikipedia2Vec, TransE, TransR, RESCAL, DISTMult, and ComplEx, which KG embedding model $KG()$ provides entity embeddings that best assists the language models? This research question aims to address **Limitation G3**, which concerns finding the most suitable entity embedding model for the check-worthy sentences and tweets task.

Moreover, from Section 4.2, the identification of check-worthy sentences/tweets can be considered either as a classification task, or instead as a ranking task (as defined by the CLEF' CheckThat! Lab organisers). Hence, in the following experiments, we provide conclusions for all RQs from both the classification and ranking perspectives. In the remainder of this section, we describe the experimental setup used to address our four research questions.

4.3.1 Dataset

All our experiments related to detecting check-worthy sentences use both the CLEF'2019 & 2020 CheckThat! datasets. The CLEF'2019 & 2020 datasets consist of transcripts of

Table 4.3: Statistics of the CLEF'2019 & 2020 CheckThat! datasets.

		Training	Testing
2019	# of debates/speeches	19	7
	# of total sentences	16,421	7,079
	# of check-worthy sentences	433	110
	% of check-worthy sentences	2.637%	2.554%
2020	# of debates/speeches	50	20
	# of total sentences	42,776	21,514
	# of check-worthy sentences	487	136
	% of check-worthy sentences	1.138%	0.632%

Table 4.4: Statistics of the CLEF'2021 check-worthiness on the tweets dataset.

	Training	Validation	Testing
# of Tweets	822	140	350
# of check-worthy tweets	290	59	19
% of check-worthy tweets	35.28%	42.14	5.43%

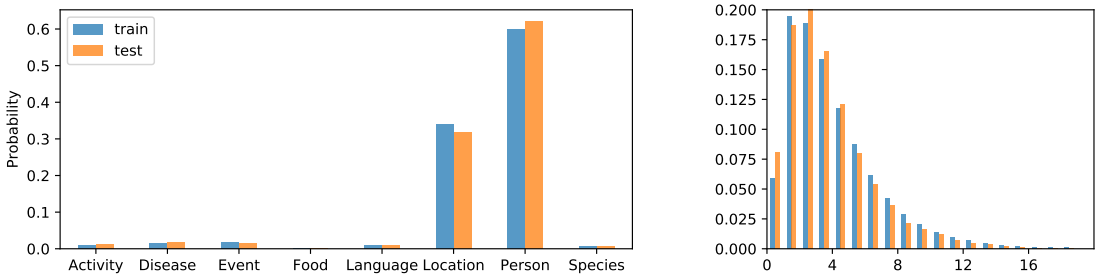
US political debates and speeches in the time period 2016-2019, collected from various news outlets[6]. Each sentence has been manually compared with `factcheck.org` by the organisers. If the sentence appeared in `factcheck.org` and is being fact-checked, it is labelled as a check-worthy claim. Table 4.5 shows examples extracted from a speech by Senator Ted Cruz. The CLEF' 2019 & 2020 CheckThat! Labs provided data splits for training and testing purposes, which we also use in this chapter. Table 4.3 shows the statistics of the training and testing sets. In particular, we observe that the prevalence of check-worthy sentences is reduced in the 2020 dataset compared to the 2019 dataset.

Our experiments related to detecting check-worthy tweets use the CLEF'2021 Task 1a English dataset. The CLEF'2021 Task 1a English dataset consists of tweets that are collected in relation to COVID19, and manually identified as either check-worthy or not check-worthy. Table 4.4 shows the statistics of the training, validation, and testing sets. In particular, we observe that the percentage of check-worthy tweets in the testing set is 15.39% of that in the training set and 12.88% of that in the validation set.

Next, Figure 4.2 shows the distribution of entity types and occurrences. Specifically, Figure 4.2a shows the proportion of each entity type appearing in the 2019 dataset[7]. In particular, it can be seen that the *Person* and *Location* types are the most commonly identified entities in the dataset, and together they account for 90% of all the entities detected. Figure 4.2b shows the number of entities appearing in each sentence. We observe that sentences with 0-2 entities account for more than 40% of the sentences, while sentences with 3 entities account for $\sim$15% of sentences. The observation of these distributions of the number of entities presented in each sentence further strengthens the reasons for using entity pairs

[6]ABC, Washington Post, CSPAN, etc. [11] are in English only.

[7]Similar distributions were observed for the 2020 and 2021 datasets, and hence are omitted.



(a) Types of entities.

(b) Number of entities per sentence.

Figure 4.2: Distribution of the entity types, and the number of entities per sentence, in the CLEF CheckThat! 2019 dataset. Entities are detected using DBpedia Spotlight. Note that we omit the figures for the 2020 and the 2021 datasets, since we observe similar distributions.

Table 4.5: A debate transcript from the CLEF'2019 CheckThat! dataset. Sentences are labelled check-worthy (1) or not (0).

Speaker	Sentence	Label
Cruz	You know, in the past couple of weeks the Wall Street Journal had a very interesting article about the state of Arizona.	0
Cruz	Arizona put in very tough laws on illegal immigration, and the result was illegal immigrants fled the state, and what's happened there – it was a very interesting article.	1
Cruz	Some of the business owners complained that the wages they had to pay workers went up, and from their perspective that was a bad thing.	0
Cruz	But, what the state of Arizona has seen is the dollars they're spending on welfare, on prisons, and education, all of those have dropped by hundreds of millions of dollars.	1

(described in Section 4.2.4).

4.3.2 Models and Baselines

In this section, we describe the tools and methods we use in our experiments, along with the baseline approaches.

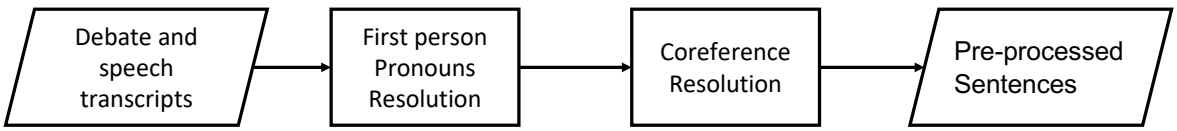


Figure 4.3: The pre-processing procedure. A parallelogram represents input and/or output; a rectangle represents a process; an arrow represents the relationship flow between two components.

4.3.2.1 Processing

American political debates usually consist of two or more participants, and one or more moderators, and each debate has different participants. In political debates, it is not explicitly apparent to the system which participants are referenced by which pronouns. Similarly, implicit pronouns can also be used to identify a specific person or a particular entity previously mentioned or known, leading to possible confusion. To combat the above mentioned challenges in implicit references, we propose a two-step procedure to resolve the implicit references found in the debates. In doing so, we aim to ensure that any implied entities in the text are therefore explicitly available for analysis by the later stages of our model (e.g. the language models and entity linking). Figure 4.3 illustrates the two steps of our preprocessing procedure: first-person pronoun resolution, and coreference resolution. Table 4.6 presents detailed examples of sentences that have gone through the preprocessing procedures. We describe the two steps procedures as follows:

1. **First-person pronouns resolution**: In this step, we simply change all the first-person pronouns in each sentence to the current speaker's name.

2. **Coreference resolution**: Coreference resolution is the task of finding the entity expression that a pronoun refers to within a piece of text. In our proposed procedure, we use coreference resolution to replace implicit mentions to one of the previously stated real-world entities. Specifically, we use the implementation of Lee et al. [92][8] of a higher-order coreference resolution method, applied to pairs of sentences. Therefore, the span of possible references for a pronoun is from either the current sentence, or the antecedent of the sentence, regardless of any change in the speaker.

Note that we do not apply the coreference resolution to the tweets datasets, as the CLEF CheckThat! 2021 tweets dataset does not consider tweet threads that may need coreference resolution.

4.3.2.2 Entity Linking

To explicitly address the entities that occur in each sentence and tweet, we deploy a named entity linking method to extract entities from each sentence and sentence. In our experiments, we use DBpedia Spotlight[9] to extract entities from each pre-processed sentence, with the confidence threshold set to 0.35[10]. We selected 0.35 as our final confidence score after fine tuning the confidence score $\in [[0.1, 0.6]$ on the training set of the CLEF 2019 dataset.

[8] https://github.com/kentonl/e2e-coref

[9] https://www.dbpedia-spotlight.org/

[10] We note that there are better performing entity linking models. However, to maintain compatibility with the previous research we only use DBpedia Spotlight as the entity linking method in this study.

Table 4.6: An example of the results of the pre-processing procedure. **Bold** denotes the pronouns that should have been changed to the entity it refers to. *Italic* denotes the changed results of the pre-processing procedure. Underline denotes the word is referring to an entity.

Speaker	Original text	after pre-processing	type of results
BLITZER	When nearly half of the delegates ..., and the biggest prize of the night is Texas.	When nearly half of the delegates ..., and the biggest prize of the night is Texas .	No entities to be resolved
BLITZER	Immigration is a key issue in **this state**, for all voters nationwide...	Immigration is a key issue in ***Texas*** , for all voters nationwide...	Correct resolution
BLITZER	So, **that**'s where we begin.	So , ***Immigration*** 's where we begin .	Correct resolution
BLITZER	Mr. Trump, you've called for a deportation force to remove the 11 million undocumented immigrants from the United States.	Mr. Trump, you've called for a deportation force to remove the 11 million undocumented immigrants from the United States...	No entities to be resolved
BLITZER	You've also promised to let what you call, "the good ones", come back in.	You 've also promised to let what you call , " the good ones " , come back in .	No entities to be resolved
BLITZER	Your words, "the good ones", after they've been deported.	Your words , " the good ones " , after they 've been deported .	No entities to be resolved
BLITZER	Senator Cruz would not allow them to come back in.	Senator Cruz would not allow ***they*** to come back in .	Incorrect resolution
BLITZER	**He** says that's the biggest difference between the two of you.	***Senator Cruz*** says that 's the biggest difference between the two of you .	Correct resolution
BLITZER	**He** calls your plan amnesty.	He calls the two of you plan amnesty.	Missing resolution

4.3.2.3 Entity Embeddings and Similarity

We use six entity embeddings methods (introduced in Section 2.7.1) to represent the head entity e_h as $\overrightarrow{e_h}$ and the tail entity e_t as $\overrightarrow{e_h}$, and combined them represent the entity pairs as $\overrightarrow{e_{pair}}$[11], as follows:

- **Wikipedia2Vec** [194] uses the extended skip-gram methods with a link-based measure [190] and an anchor context model to learn the embeddings of entities.
- **TransE** [16] aims to embed a triplet $e = \langle e_h, relation, e_t \rangle$ into the same lower dimensional space, where $\overrightarrow{e_h} + \overrightarrow{r}$ should result in $\overrightarrow{e_t}$.

[11] We acknowledge that these entity pair representations can also be used to calculate their similarities. However our preliminary experiments shows that such methods produce less than satisfactory results, thus we omit such setup.

- **TransR** [100] is built upon TransE, where the relation embedding is projected into a separate relation space, in order to more accurately represent the rich and diverse information between entities and relations.

- **RESCAL** [129] uses a three-way tensor learning method to model the triplet of $e = \langle e_h, relation, e_t \rangle$, for a more flexible representation of the relationship and entities.

- **DISTMult** [195] uses a single vector to represent both entities and the relation by simplifying the bi-linear interaction between the entity and the relation, where the relation vector is represented using the diagonal matrix of the interaction.

- **ComplEx** [180] uses complex embeddings and the Hermitian dot product to represent the relation between two entities, and yields a better performance than its predecessors (e.g., TransE, TransR, RESCAL) on the entity-linking task [180].

For the TransE, TransR, RESCAL, DistMult and ComplEx models, we use triplets extracted from Freebase (FB15K) [16] as training data. These models are trained using code provided by Zheng et al. [205][12]. For the Wikipedia2Vec model we use the pre-trained model provided by the author[13].

To calculate the entity similarity, we use the Sematch [208][14]'s KG semantic similarity and relatedness algorithms, based on the algorithm proposed by Zhu and Iglesias [209], to calculate the similarity and relatedness of an entity pair of entities appearing in each sentence.

4.3.2.4 Language Representations

We use six different text representation models $LM()$ (introduced in Section 2.6.2) to represent each sentence and/or tweets[15], as follows:

- **TF.IDF** is a commonly used BoW model to represent the text based on the word frequencies. We include TF.IDF as a baseline. We use sci-kit learn implemented TF.IDF model[16], with English stop words removed, and maximum features as 5000.

- **BiLSTM+attention** (denoted as BiLSTM+att) is widely used in the literature to learn a language model from the training data. It appeared in several solutions [69], which were deployed in the CLEF'2019 CheckThat! Lab, where it was shown that an LSTM-based

[12] https://github.com/awslabs/dgl-ke

[13] https://wikipedia2vec.github.io/wikipedia2vec/pre-trained/

[14] https://github.com/gsi-upm/sematch

[15] all models except BERTweet model are used for both sentences and tweets, BERTweet model is exclusively used for tweets)

[16] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

language model can effectively represent the sentences in the check-worthiness identification task. Thus, in this chapter, we use BiLSTM+att as a non-pre-trained language model baseline, in order to obtain a fair comparison among all the language representation methods.

- **BERT** [40] is a pre-trained language model that has been shown to be effective in many information retrieval and natural language processing tasks [110, 167, 198]. In this study, we are also interested in determining if the BERT model also performs well on the very specific task of check-worthy sentence identification, or if it can be enhanced by supplementary information such as entity embeddings (as discussed in the next section).

- **ALBERT** [89] is a derivative of the original BERT model that aims to reduce the number of parameters. Specifically, ALBERT uses a factorised embedding parameterisation method to decompose the vocabulary size and the hidden layer size, by projecting the vocabulary twice rather than once. Moreover, cross-layer parameter sharing and inter-sentence coherence loss are used to further reduce the need of parameters updating. ALBERT achieved a new SOTA performance with fewer parameters and shorter training time on SQuAD and MNLI datasets, compared to the original BERT model [89].

- **RoBERTa** [106] aims to improve over BERT by training the model for more iterations, using longer sentence sequences, with bigger batches over more data. RoBERTa also removes the next sentence prediction objective in training. Similar to the ALBERT model, RoBERTa results in improved performance over the standard BERT model [106].

- **BERTweet** [126] is a pre-trained language model that deploys RoBERTA architecture on 850M English tweets. The main objective of BERTweet is to build a pre-trained language model specifically for analysing tweets. We use BERTweet as a language model only for the CLEF CheckThat! 2021 dataset, where the task is to identify the most check-worthy tweets.

We use the HuggingFace language model implementations [191][17]. Specifically, we use the BERT-Cased English model (12-layer, 768-hidden, 12-heads, 110M parameters); the Albert-base-v2 English model (12-layer, 128-hidden, 12-heads, 1M parameters); the RoBERTa-base English model (12-layer, 768-hidden, 12-heads, 125M parameters); and the BERTweet-base model (12-layer, 768-hidden, 12-heads, 135M parameters). We fine-tune all the BERT-related language models on the training datasets as presented in Section 4.3.1. All other parameters remain at their recommended settings.

[17] `https://github.com/huggingface/transformers`

4.3.2.5 Baselines

We compare our generated Entity-Assisted models to the following baselines:

- **Random classifier**: We apply a random classifier using the stratified strategy, as the weakest baseline.
- **SVM(TF.IDF**: We apply an SVM text classifier using TF.IDF features, as a weak baseline using the traditional text representation methods and the statistical machine learning model. We select our hyperparameters by applying cross-validation on the training data. Specifically, we use the sci-kit learn SVM implementation with an RBF kernel, a C penalty of 10, and a γ of 0.1 in our trained SVM classifier. We use class weights based on the training data to prevent the imbalanced data from compromising our experimental results. For the classification task we obtain the predicted class label for each sentence from $f^{cls}()$ (as per Equation (4.8)), while for the ranking task we obtain a score for each sentence in the range (0, 1) from $f^{rank}()$ (as per Equation (4.9)). We use the same SVM settings for SVM(TF.IDF) **similarity** (introduced below).
- **SVM(TF.IDF) + Entity Similarity and Relatedness**: As a baseline method, we append two graph-based entity similarity and relatedness scores – obtained using Sematch [208], to the TF.IDF feature vectors of the SVM model. We introduce this baseline to compare the use of entity similarity and relatedness scores with using embedded entity vectors in identifying check-worthy sentences and tweets.
- **BiLSTM + Att**: We deploy a BiLSTM + Att model (100 hidden units) with an attention mechanism, implemented using Tensorflow. We initialise the embedding layer of BiLSTM using the pre-trained GloVe embeddings (300 dimensions). We introduce this baseline to compare the performances between pre-trained deep learning language model with locally trained deep learning language model.
- **CLEF'2019, 2020 & 2021 CheckThat! Lab leaderboards**: For the ranking task, we additionally compare the performances of our models with the runs of the top three groups on the official CLEF'2019, 2020 & 2021 leaderboards[18].

4.3.3 Evaluation Metrics

For evaluating the classification task, we use the standard classification metrics (Precision, Recall, F1). Significant differences are measured using the McNemar's test.[19] On the other

[18]From `https://github.com/apepa/clef2019-factchecking-task1` for the 2019 results, `https://github.com/sshaar/clef2020-factchecking-task5` for the 2020 results, and the overview paper [124] for the 2021 results.

[19]We evaluate the classification task only using the CLEF'2019 CheckThat! Lab dataset, and 2021 Tweet dataset, as our prior results found that it is not possible to derive meaningful results from the 2020 dataset, due

hand, for evaluating the ranking effectiveness, we apply the ranking metrics used by the CheckThat! Lab organisers, namely Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Mean Precision at rank k (P@k, k=$\{$1,5,10,20,50$\}$) [20]. Means are calculated over the seven and twenty debates and speeches (equivalent to 7 and 20 queries) in the CheckThat! 2019 and 2020 test sets, respectively - therefore, due to the small number of rankings being evaluated, significance testing is not meaningful. Finally, note that it is not possible to evaluate the CLEF'2019 & 2020 CheckThat! Lab participants' approaches using the classification metrics – this is because the participants' runs have scores rather than predicted labels, and do not contain predictions for all sentences in the dataset. Moreover, it is also not possible to combine our approach with the participants' runs, since we do not have the predicted scores of the participants' runs on the training sets.

Table 4.7: Classification performances on the CheckThat! 2019 and 2021 datasets, alternating language models $LM()$ only. **Bold** indicates the best performance in the respective dataset; Numbers in the Significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	$LM()$	P	R	F1	Significance
CLEF'2019 CheckThat! results					
1	Random Classifier	0.01	0.01	0.01	-
2	SVM(TF.IDF)	0.01	0.01	0.01	-
3	BiLSTM+att	0.12	0.07	0.09	1,2
4	BERT	0.12	0.09	0.10	1-3
5	ALBERT	**0.14**	**0.11**	**0.12**	1-4
6	RoBERTa	**0.14**	**0.11**	0.11	1-4
CLEF'2021 CheckThat! results					
7	Random Classifier	0.05	0.05	0.05	-
8	SVM(TF.IDF)	0.05	0.11	0.07	7
9	BiLSTM+att	0.05	0.11	0.07	7
10	BERT	0.08	0.16	0.10	7-9
11	ALBERT	0.08	0.16	0.11	7-9
12	RoBERTa	0.09	0.16	0.11	7-9
13	BERTweet	**0.16**	**0.47**	**0.23**	7-12

4.4 Experimental Results

In this section, we present the results of the experiments that address RQs 4.1 - 4.4. In particular, for both the check-worthy sentence classification and ranking tasks, Sections 4.4.1 - 4.4.4 respectively address: the effectiveness of the BERT-related language models $LM()$

to the small number of positive data in the test set.

[20]For the CLEF'2021 CheckThat! Lab Tweet dataset, we use MAP, MRR for consistency, although the ranking task is to rank all check-worthy tweet in a single run, equivalent to a single query retrieval task.

Table 4.8: Ranking performances on the CheckThat! 2019, 2020, and 2021 dataset, alternating language models $LM()$ only. **Bold** indicates the best performance in each group.

#	$LM()$	MAP	MRR	P@1	P@5	P@10	P@20	P@50
CLEF'2019 CheckThat! Experimental results								
1	SVM(TF.IDF)	0.1193	0.3513	0.1429	**0.2571**	**0.1571**	0.1714	0.1086
2	BiLSTM+att	**0.1453**	0.2432	0.1429	0.1429	0.1429	**0.1857**	**0.1343**
3	BERT	0.0715	0.2257	0.1429	0.2000	0.1286	0.0857	0.0600
4	ALBERT	0.1332	**0.4176**	**0.3098**	0.2000	0.1429	0.1286	0.0929
5	RoBERTa	0.1011	0.3158	0.2286	0.2000	0.1429	0.1286	0.0929
CLEF'2019 CheckThat! Submitted Runs								
6	Copenhagen-primary	0.1660	0.4176	**0.2857**	**0.2571**	0.2286	0.1571	0.1229
7	Copenhagen-contr.-1	0.1496	0.3098	0.1429	0.2000	0.2000	0.1429	0.1143
8	Copenhagen-contr.-2	0.1580	0.2740	0.1429	0.2286	**0.2429**	0.1786	0.1200
9	TheEarthIsFlat-primary	0.1597	0.1953	0.0000	0.2286	0.2143	0.1857	**0.1457**
10	TheEarthIsFlat-contr.-1	0.1453	0.3158	**0.2857**	0.1429	0.1429	0.1357	0.1171
11	TheEarthIsFlat-contr.-2	**0.1821**	**0.4187**	**0.2857**	0.2286	0.2286	**0.2143**	0.1400
12	IPIPAN-primary	0.1332	0.2865	0.1429	0.1430	0.1715	0.1500	0.1171
CLEF'2020 CheckThat! Experimental results								
13	SVM(TF.IDF)	**0.0946**	0.1531	0.0000	**0.0600**	0.0400	**0.0450**	0.0240
14	BiLSTM+att	0.0151	0.0320	0.0000	0.0100	0.0150	0.0075	0.0090
15	BERT	0.0262	0.0819	0.0500	0.0300	0.0250	0.0125	0.0110
16	ALBERT	0.0537	**0.2145**	**0.2000**	0.0800	**0.0500**	0.0250	**0.1600**
17	RoBERTa	0.0424	0.1315	0.1000	0.0600	0.0400	0.0200	0.1400
CLEF'2020 CheckThat! Submitted Runs								
18	NLP_IR@UNED-primary	**0.0867**	**0.2770**	**0.1500**	**0.1300**	**0.0950**	**0.0725**	**0.0390**
19	NLP_IR@UNED-contr.-1	0.0849	0.2590	**0.1500**	0.1200	0.0900	0.0675	0.0370
20	NLP_IR@UNED-contr.-2	0.0408	0.1170	0.0500	0.0700	0.0450	0.0275	0.0180
21	UAICS-primary	0.0515	0.2247	**0.1500**	0.0700	0.0500	0.0375	0.0270
22	UAICS-contr.-1	0.0431	0.1735	0.1000	0.0500	0.0550	0.0450	0.0250
23	UAICS-contr.-2	0.0328	0.1138	0.0500	0.0300	0.0350	0.0175	0.0190
24	TobbEtuP-primary	0.0183	0.0326	0.0000	0.0200	0.0100	0.0100	0.0060
25	TobbEtuP-contr.-1	0.0417	0.0784	0.0500	0.0300	0.0150	0.0150	0.0180
CLEF'2021 CheckThat! Experimental results								
26	SVM(TF.IDF)	0.0608	0.1111	0.0000	0.0000	0.1000	0.0500	0.0400
27	BiLSTM+att	0.0635	0.1429	0.0000	0.0100	0.1000	0.0500	0.0400
28	BERT	0.0757	0.1429	0.0000	0.0000	0.1000	0.1000	0.0600
29	ALBERT	0.0777	0.1429	0.0000	0.0000	0.1000	0.1000	0.0600
30	RoBERTa	0.0806	0.1429	0.0000	0.0000	0.2000	0.1000	0.0600
31	BERTweet	0.1326	0.5000	0.0000	0.2000	0.1000	0.1500	**0.1800**
CLEF'2021 CheckThat! Submitted Runs								
32	NLP&IR@UNED	**0.2240**	**1.0000**	**1.000**	**0.4000**	0.3000	0.2000	0.1600
33	Fight for 4230	0.1950	0.3333	0.000	**0.4000**	**0.4000**	**0.2500**	0.1600
34	UPV	0.1490	**1.0000**	**1.000**	0.2000	0.2000	0.1000	0.1200

(presented in Section 4.3.2.4); the usefulness of entity embeddings; the most effective combination method $COM()$ for representing entity pairs (presented in Equation (4.8)); and the most effective KG embedding model $KG()$ from which to obtain the entity embeddings (presented in Section 4.3.2.3).

4.4.1 RQ 4.1: BERT-related Language Models vs. Baselines

RQ 4.1 answers the research question that whether BERT-related language models outperform the TF.IDF and BiLSTM baselines in identifying check-worthy sentences/tweets. RQ 4.1 to address the **Limitation L1** presented in Section 2.6.4, which concerns finding the best language models for the identification of check-worthy claims. Table 4.7 presents classification results on CLEF CheckThat! 2019 & 2021 datasets, and Table 4.8 presents the baselines, and the attained ranking performances using the language models only, on the CLEF CheckThat! 2019, 2020 & 2021 datasets.

We firstly consider Table 4.7, which reports the attained precision, recall, and F1 scores when treating check-worthy sentence identification as a classification task, on the CLEF CheckThat! 2019 and 2021 datasets. Firstly, in terms of F1 on the 2019 dataset, we note the relative weak performance of a classical SVM classifier with TF.IDF features (row 2), which performs equivalently to a random classifier. Indeed, while the SVM classifier has been trained using class weights to alleviate the issue of class imbalance, the low performance of SVM illustrates the difficulty of this task, and underlines that simply matching on *what is being said* by the speakers is insufficient to attain high accuracies on this task. Next, the BiLSTM+att classifier (row 3) markedly outperforms the random classifier, demonstrating that the deployment of pre-trained (i.e., GloVe) word embeddings allows a more flexible classifier not tied to the exact matching of tokens. Moreover, the use of the attention mechanism in BiLSTM also emphasises the importance of the context of each word. Finally, the state-of-the-art BERT-related models (BERT model, row 4; ALBERT model, row 5, and RoBERTa model, row 6) significantly (McNemar's Test, $p<0.01$) outperform the random classifier, the SVM classifiers, and the BiLSTM+att classifiers. Thus, we conclude that, when treating the task as a classification task, all of the BERT-related language models can significantly outperform the SVM and BiLSTM+att classifiers. Among all the BERT-related models, ALBERT exhibits the highest performance. This is expected from the literature, as ALBERT outperforms all other tested language models on a range of benchmarks [89], such as GLUE, RACE, and SQuAD.

Similarly, when tested on the CLEF CheckThat! 2021 tweets dataset, SVM(TF.IDF), BiLSTM, BERT, ALBERT, and RoBERTa all perform significantly better than the random classifier, but remain relatively ineffective. Indeed, the BERTweet model (row 13) performs significantly and markedly better than all the other models, while more than doubling the F1 score of the next best score (0.11 from row 11 and 12). Thus, we conclude that the BERTweet model, a language model trained on tweets, indeed outperforms all the other traditional and BERT related models, and exhibits the best performance, in identifying the most check-worthy tweets.

Moving next to the ranking task on the 2019 dataset, Table 4.8 shows that the BERT model

(row 3) underperforms the classical SVM classifier using TF.IDF features (row 1). Both ALBERT (row 4) and RoBERTa (row 5) outperform SVM(TF.IDF) and BiLSTM+att (rows 1, 2) in terms of MRR. However, in terms of MAP, both ALBERT and RoBERTa only outperform SVM(TF.IDF) (row 1), and still underperform compared to BiLSTM+att (row 2). Next, when considering the results of the ranking task on the 2020 dataset, BERT (row 15), ALBERT (row 16) and RoBERTa (row 17) models all outperform BiLSTM+att (row 14) and SVM(TF.IDF) (row 13) on both MAP and MRR. Similar to that of the ranking task on the 2021 dataset, all the BERT related models (BERT, row 28; ALBERT, row 29; RoBERTa, row 30, and BERTweet, row 31) outperform BiLSTM+att (row 27) and SVM(TF.IDF) (row 26) on both MAP and MRR.

While the contrast between the F1 classification and the ranking results on the 2019 dataset is notable, the low classification recall for all models suggests that BERT, ALBERT, and RoBERTa (c.f. rows 4, 5, 6 in Table 4.7) cannot retrieve the most difficult check-worthy sentences among the 2019 dataset, and hence also exhibit low MAP performances in the ranking task. However, we observe that BERT-related language models indeed outperform SVM(TF.IDF) and BiLSTM+att in both classification (rows 10-13 vs. rows 8, 9 in Table 4.7) and ranking on the 2021 dataset (rows 28-31 vs. rows 26,27 in Table 4.8), especially when using the BERTweet language model. This suggests that the BERTweet language model is well suited to identify check-worthy tweets from the 2021 Tweets dataset. From Table 4.21 we observe the inconsistent performances for the same language model across the 2019 and 2020 ranking datasets (i.e., row 1 vs. 16, row 4 vs. 18, row 7 vs. 20, row 10 vs. 22, row 13 vs. 24). We postulate that this may be caused by the markedly different proportion of positive examples in the two test sets (as illustrated by the percentage of the check-worthy sentences in Table 4.3).

Overall, in answer to RQ 4.1, we conclude that the BERT, ALBERT, RoBERTa, and BERTweet models perform well at classifying and ranking check-worthy sentences/tweets. Specifically, BERT-related models are most effective at higher rank sentences/tweets. On both tasks, ALBERT performs the best among the BERT-related language models at classifying and ranking the most check-worthy sentences, while BERTweet performs the best at classifying and ranking the most check-worthy tweets.

4.4.2 RQ 4.2: Using Entity Embeddings

RQ 4.2 aims to answer the question that whether combing entity embeddings with language models can improve the classification and ranking performance on identifying check-worthy tweets and sentences, than using language models alone. This research question addresses **Limitations L2 & G1** discussed in Section 2.6.4 and Section 2.7.1.4 respectively, which concern enriching language models with additional embedded entity information.

Table 4.9: Classification performances on the CheckThat! 2019 dataset, alternating language models $LM()$ and entity embedding models $KG()$, and entity representation combination models $COM()$. **Bold** indicates the best performance; Numbers in the Significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	$LM()$	$KG()$	$COM()$	P	R	F1	Significance
1	SVM(TF.IDF)	-	-	0.01	0.01	0.01	-
2	SVM(TF.IDF)	Wikipedia2Vec	similarity()	0.04	0.03	0.03	1
3	SVM(TF.IDF)	Wikipedia2Vec	emb_concat()	0.06	0.05	0.05	1,2
4	SVM(TF.IDF)	Wikipedia2Vec	emb_prod()	0.05	0.04	0.04	1,2
5	BiLSTM+att	-	-	0.12	0.07	0.09	1-4
6	BiLSTM+att	Wikipedia2Vec	similarity()	0.12	0.08	0.1	1-5
7	BiLSTM+att	Wikipedia2Vec	emb_concat()	0.13	0.1	0.11	1-6
8	BiLSTM+att	Wikipedia2Vec	emb_prod()	0.12	0.09	0.1	1-5
9	BERT	-	-	0.12	0.09	0.1	1-5
10	BERT	Wikipedia2Vec	similarity()	0.12	0.1	0.11	1-6
11	BERT	Wikipedia2Vec	emb_concat()	0.19	0.11	0.14	1-10, 13
12	BERT	Wikipedia2Vec	emb_prod()	0.18	0.11	0.13	1-10, 13
13	ALBERT	-	-	0.14	0.11	0.12	1-10
14	ALBERT	Wikipedia2Vec	similarity()	0.14	0.14	0.14	1-10, 13
15	ALBERT	Wikipedia2Vec	emb_concat()	**0.22**	**0.15**	**0.18**	1-14, 17-20
16	ALBERT	Wikipedia2Vec	emb_prod()	0.20	0.14	0.16	1-14, 17, 18
17	RoBERTa	-	-	0.14	0.11	0.12	1-10
18	RoBERTa	Wikipedia2Vec	similarity()	0.14	0.13	0.13	1-10
19	RoBERTa	Wikipedia2Vec	emb_concat()	0.21	**0.15**	0.17	1-14, 17, 18
20	RoBERTa	Wikipedia2Vec	emb_prod()	0.19	0.14	0.16	1-14, 17, 18

To address RQ 4.2 for the classification task, Table 4.9 presents the results obtained on the CLEF CheckThat! 2019 dataset, and Table 4.10 presents the results on the 2021 tweets dataset. For the ranking task, Tables 4.11 and 4.12 present the results obtained from the CLEF CheckThat! 2019 dataset, and the 2021 tweet dataset, respectively.

Firstly, from Table 4.9, we note that the F1 performance of the SVM classifier is improved by adding the entity similarity scores using similarity() (row 2 vs row 1). Similarly, concatenating entity representation $\overrightarrow{e_{pair}}$ (either by $\overrightarrow{e_{pair}}$ = emb_prod($\overrightarrow{e_h}, \overrightarrow{e_t}$), or by $\overrightarrow{e_{pair}}$ = emb_concat($\overrightarrow{e_h}, \overrightarrow{e_t}$), presented in Secion 4.2.2) with language representation $\overrightarrow{x}$ also improves the SVM classifier's performance (rows 3, 4 vs. row 1), using the 2019 dataset, in terms of precision, recall and F1 compared to SVM(TF.IDF) without entity information. Next, we observe that all of the neural language models (i.e., BiLSTM+att, BERT, ALBERT, RoBERTa) also exhibit a significantly improved F1 when combined with entity embeddings (rows 7 & 8 vs. 5; rows 11 & 12 vs. 9; rows 15 & 16 vs. 13; rows 19 & 20 vs 17). On the contrary, even though we do observe an improvement when the neural models are combined with entity similarities (row 6 vs. 5; row 10 vs. 9, row 14 vs. 13; row 18 vs. 17), the improvement is not significant. Moreover, combining the neural models with the entity embedding

Table 4.10: Classification performances on the CheckThat! 2021 Tweets dataset, alternating language models $LM()$ and entity embedding models $KG()$, and entity representation combination models $COM()$. **Bold** indicates the best performance; Numbers in the Significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	$LM()$	$KG()$	$COM()$	P	R	F1	Significance
1	SVM(TF.IDF)	-	-	0.05	0.11	0.07	-
2	SVM(TF.IDF)	Wikipedia2Vec	similarity()	0.05	0.11	0.07	-
3	SVM(TF.IDF)	Wikipedia2Vec	emb_concat()	0.08	0.16	0.10	1,2,4,5
4	SVM(TF.IDF)	Wikipedia2Vec	emb_prod()	0.05	0.11	0.007	-
5	BiLSTM+att	-	-	0.05	0.11	0.07	-
6	BiLSTM+att	Wikipedia2Vec	similarity()	0.10	0.21	0.13	1-5, 9,13,17
7	BiLSTM+att	Wikipedia2Vec	emb_concat()	0.10	0.21	0.14	1-5, 9,13,17
8	BiLSTM+att	Wikipedia2Vec	emb_prod()	0.10	0.21	0.14	1-5, 9,13,17
9	BERT	-	-	0.08	0.16	0.10	1,2,4,5
10	BERT	Wikipedia2Vec	similarity()	0.10	0.21	0.13	1-5, 9,13,17
11	BERT	Wikipedia2Vec	emb_concat()	0.11	0.21	0.14	1-6, 9,10,13,17
12	BERT	Wikipedia2Vec	emb_prod()	0.11	0.21	0.14	1-6, 9,10,13,17
13	ALBERT	-	-	0.08	0.16	0.11	1-5
14	ALBERT	Wikipedia2Vec	similarity()	0.11	0.21	0.14	1-6, 9,10,13,17
15	ALBERT	Wikipedia2Vec	emb_concat()	0.11	0.21	0.14	1-6, 9,10,13,17
16	ALBERT	Wikipedia2Vec	emb_prod()	0.11	0.21	0.14	1-6, 9,10,13,17
17	RoBERTa	-	-	0.09	0.16	0.11	1-5
18	RoBERTa	Wikipedia2Vec	similarity()	0.11	0.21	0.14	1-6, 9,10,13,17
19	RoBERTa	Wikipedia2Vec	emb_concat()	0.11	0.21	0.14	1-6, 9,10,13,17
20	RoBERTa	Wikipedia2Vec	emb_prod()	0.11	0.21	0.14	1-6, 9,10,13,17
21	BERTweet	-	-	0.16	0.47	0.23	1-20
22	BERTweet	Wikipedia2Vec	similarity()	0.16	0.53	0.25	1-21
23	BERTweet	Wikipedia2Vec	emb_concat()	**0.18**	**0.58**	**0.27**	1-22,24
24	BERTweet	Wikipedia2Vec	emb_prod()	0.16	0.53	0.25	1-21

information (using either of the entity combination methods) significantly outperforms the corresponding language model combined with entity similarity scores. Indeed, our proposed entity-assisted ALBERT classifier using the emb_concat() method (row 16) attains the highest overall classification performance (an F1 score of 0.18). Table 4.9 further shows that almost all neural models with all types of entity embeddings outperform the corresponding language models alone, in terms of F1.

Table 4.10 shows similar results on classifying check-worthy tweets. Specifically, the F1 performance of all language models improved when the language models are combined with entity representations (rows 2-4 vs. 1; rows 6-8 vs. 5; rows 10-12 vs. 9; rows 14-16 vs. 13; rows 18-20 vs. 17; and rows 22-24 vs. 21). We note that representing entities as embeddings improve upon using the similarity scores significantly only when the language models are one of the SVM(TF.IDF) (row 3 vs. 2), BERT (row 11 vs. 10), and BERTweet (row 23 vs. 22) models. Thus, we conclude that the entity information can indeed improve the classification

performance in the identification of check-worthy sentences.

Turning to the ranking task, in Table 4.11, we observe that the use of entities (i.e., similarity(), emb_concat(), and emb_prod(), presented in Section 4.2.2 and Equation (4.5)), enhances most of the approaches: the effectiveness of the SVM(TF.IDF) model is enhanced on MAP, P@1, and P@10. On the other hand, while BiLSTM+att is enhanced for MRR and P@1, when combined with , the MAP performances are damaged by the any type of entity information (rows 6-8 vs. 5). Finally, the BERT-related models (i.e., BERT, ALBERT, RoBERTa) are enhanced by all three types of entity information, regardless of the entity embedding combination model used, in terms of MAP, MRR, and P@1 (rows 10-12 vs. 9; rows 14-16 vs. 13; rows 18-20 vs. 17). When tested on the 2021 tweets dataset, Table 4.12 shows that all types of entities information enhanced the language models' performance, regardless of the entity combination model, while BERTweet together with the emb_concat() method obtained the best performance on all metrics. Thus, we conclude that entity embeddings can consistently enhance the BiLSTM+att models for ranking sentences and tweets on high precision metrics such as MRR and P@1, as well as enhance the SVM(TF.IDF) and neural language models (i.e., BERT, ALBERT, RoBERTa, and BERTweet) across all the evaluation metrics.

Therefore, in response to RQ 4.2, we conclude that using entity embeddings – regardless of the KG embedding model – does help to improve the BERT-related language models' performance, on both precision and recall for the classification task, and on MAP, MRR and P@1 for the ranking tasks.

Table 4.11: Ranking performances on the CheckThat! 2019 dataset, alternating language models $LM()$ and entity embedding models $KG()$, and entity representation combination models $COM()$. **Bold** indicates the best performance.

#	$LM()$	$KG()$	$COM()$	MAP	MRR	P@1	P@5	P@10	P@20	P@50
CLEF'2019 CheckThat! Experimental results										
1	SVM(TF.IDF)	-	-	0.1193	0.3513	0.1429	0.2571	0.1571	0.1714	0.1086
2	SVM(TF.IDF)	Wikepedia2Vec	similarity()	0.1263	0.3254	0.2857	0.2000	0.2000	0.1286	0.0915
3	SVM(TF.IDF)	Wikepedia2Vec	emb_concat()	0.1332	0.3361	0.3254	0.2000	0.2000	0.1286	0.0915
4	SVM(TF.IDF)	Wikepedia2Vec	emb_prod()	0.1332	0.3361	0.3254	0.2000	0.2000	0.1286	0.0915
5	BiLSTM+att	-	-	0.1453	0.2432	0.1429	0.1429	0.1429	0.1857	0.1343
6	BiLSTM+att	Wikepedia2Vec	similarity()	0.0715	0.2857	0.2432	0.1429	0.1286	0.0714	0.0314
7	BiLSTM+att	Wikepedia2Vec	emb_concat()	0.0659	0.3361	0.2857	0.1429	0.1429	0.0714	0.0314
8	BiLSTM+att	Wikepedia2Vec	emb_prod()	0.0659	0.3158	0.2000	0.1429	0.1286	0.0714	0.0714
9	BERT	-	-	0.0715	0.2257	0.1429	0.2000	0.1286	0.0857	0.0600
10	BERT	Wikepedia2Vec	similarity()	0.0826	0.3158	0.3098	0.2000	0.1286	0.0929	0.0600
11	BERT	Wikepedia2Vec	emb_concat()	0.1011	**0.6196**	**0.3361**	0.1714	0.1429	0.0929	0.0686
12	BERT	Wikepedia2Vec	emb_prod()	0.0826	0.3361	**0.3361**	0.1429	0.1429	0.0929	0.0929
13	ALBERT	-	-	0.1332	0.4176	0.3098	0.2000	0.1429	0.1286	0.0929
14	ALBERT	Wikepedia2Vec	similarity()	0.1453	0.4176	**0.3361**	0.2286	0.2000	0.1286	0.1286
15	ALBERT	Wikepedia2Vec	emb_concat()	**0.1580**	**0.6196**	0.3098	0.2857	**0.2571**	**0.2286**	**0.2286**
16	ALBERT	Wikepedia2Vec	emb_prod()	0.1332	0.4187	**0.3361**	0.2571	**0.2571**	0.2000	0.1286
17	RoBERTa	-	-	0.1011	0.3158	0.2286	0.2000	0.1429	0.1286	0.0929
18	RoBERTa	Wikepedia2Vec	similarity()	0.1263	0.4176	**0.3361**	0.2286	0.2000	0.1286	0.0929
19	RoBERTa	Wikepedia2Vec	emb_concat()	0.1453	0.4176	**0.3361**	**0.2857**	**0.2571**	0.2000	**0.2286**
20	RoBERTa	Wikepedia2Vec	emb_prod()	0.1332	0.4187	**0.3361**	0.2571	0.2000	0.2000	0.1286

Table 4.12: Ranking performances on the CheckThat! 2021 Tweets dataset, alternating language models $LM()$ and entity embedding models $KG()$, and entity representation combination models $COM()$. **Bold** indicates the best performance.

#	$LM()$	$KG()$	$COM()$	MAP	MRR	P@1	P@5	P@10	P@20	P@50
CLEF'2019 CheckThat! Experimental results										
1	SVM(TF.IDF)	-	-	0.0608	0.1111	0.0000	0.0000	0.1000	0.0500	0.0400
2	SVM(TF.IDF)	Wikepedia2Vec	similarity()	0.0658	0.1429	0.0000	0.0000	0.1000	0.0500	0.0400
3	SVM(TF.IDF)	Wikepedia2Vec	emb_concat()	0.0635	0.1429	0.0000	0.0100	0.1000	0.0500	0.0400
4	SVM(TF.IDF)	Wikepedia2Vec	emb_prod()	0.0846	0.5000	0.0000	0.2000	0.1000	0.0500	0.0400
5	BiLSTM+att	-	-	0.0635	0.1429	0.0000	0.0100	0.1000	0.0500	0.0400
6	BiLSTM+att	Wikepedia2Vec	similarity()	0.1402	0.5000	0.0000	0.4000	0.3000	0.1500	0.0800
7	BiLSTM+att	Wikepedia2Vec	emb_concat()	0.1149	0.5000	0.0000	0.2000	0.2000	0.1500	0.0800
8	BiLSTM+att	Wikepedia2Vec	emb_prod()	0.1147	0.5000	0.0000	0.2000	0.2000	0.1500	0.0800
9	BERT	-	-	0.0757	0.1429	0.0000	0.0000	0.1000	0.1000	0.6000
10	BERT	Wikepedia2Vec	similarity()	0.1317	0.5000	0.0000	0.4000	0.3000	0.1500	0.0800
11	BERT	Wikepedia2Vec	emb_concat()	0.1813	**1.0000**	**1.0000**	0.6000	0.3000	0.1500	0.0800
12	BERT	Wikepedia2Vec	emb_prod()	0.1550	0.5000	0.0000	0.6000	0.3000	0.1500	0.0800
13	ALBERT	-	-	0.0777	0.1429	0.0000	0.0000	0.1000	0.1000	0.0600
14	ALBERT	Wikepedia2Vec	similarity()	0.1556	0.5000	0.0000	0.6000	0.3000	0.1500	0.0800
15	ALBERT	Wikepedia2Vec	emb_concat()	0.1823	**1.0000**	**1.0000**	0.6000	0.3000	0.1500	0.0800
16	ALBERT	Wikepedia2Vec	emb_prod()	0.1816	**1.0000**	**1.0000**	0.6000	0.3000	0.1500	0.0800
17	RoBERTa	-	-	0.0806	0.1429	0.0000	0.0000	0.2000	0.1000	0.0600
18	RoBERTa	Wikepedia2Vec	similarity()	0.1638	**1.0000**	**1.0000**	0.0400	0.3000	0.1500	0.0800
19	RoBERTa	Wikepedia2Vec	emb_concat()	0.2182	**1.0000**	**1.0000**	0.0600	0.4000	0.2000	0.0800
20	RoBERTa	Wikepedia2Vec	emb_prod()	0.1997	**1.0000**	**1.0000**	0.0600	0.3000	0.1500	0.0800
21	BERTweet	-	-	0.1326	0.5000	0.0000	0.2000	0.1000	0.1500	0.1800
22	BERTweet	Wikepedia2Vec	similarity()	0.3124	**1.0000**	**1.0000**	0.6000	0.3000	0.3000	0.2000
23	BERTweet	Wikepedia2Vec	emb_concat()	**0.3268**	**1.0000**	**1.0000**	**0.8000**	**0.4000**	**0.3500**	**0.2200**
24	BERTweet	Wikepedia2Vec	emb_prod()	0.2495	0.5000	0.0000	0.6000	0.3000	0.3000	0.2000

4.4.3 RQ 4.3: Entity Representation

RQ 4.3 identifies the most effective combination method, $COM() \in \{\text{emb\_prod}, \text{emb\_concat}\}$, in improving the performance of text representations at identifying check-worthy sentences/tweets. This research question addresses **Limitation G2**, which concerns tailoring entity embeddings to better assist language models. To address RQ 4.3, we use Table 4.9 and Table 4.10 for the classification task, and Table 4.11 and 4.12 for the ranking task.

When considering identifying check-worthy sentences as a classification task, Table 4.9 shows that all of the SVM(TF.IDF) and BERT-related language models are significantly improved when combined with entity embeddings, over the language models alone or with entity similarities. Similarly, on the identifying check-worthy tweets task, Table 4.10 shows that all language models are significantly improved when combined with entity embeddings, while emb_concat() only significantly outperforms the similarity() model on SVM(TF.IDF) model, BERT, and Tweet model. Meanwhile, from both Table 4.9 and 4.10, we observe that using emb_concat() only marginally outperforms emb_prod(), without significant differences (row 3 vs. 4, row 7 vs. 8, row 11 vs. 12, row 15 vs. 16, row 19 vs. 20 in Table 4.9; row 3 vs. 4, row 7 vs. 8, row 11 vs. 12, row 15 vs. 16, row 19 vs. 20, and row 23 vs. 24 in Table 4.10).

Next, when considering the ranking task, Table 4.11 shows that the BiLSTM+att, and BERT-

related languages models all exhibit improved MRR and P@1 when combined with entity embeddings using the concatenation method, outperforming the entity similarity method using similarity() (rows 3 & 4 vs. 2; 7 & 8 vs. 6; 11 & 12 vs. 10; rows 15 & 16 vs. 14; rows 19 & 20 vs. row 18). In terms of the entity representation methods for the embedded entities, emb_concat() and emb_prod() perform similarly for SVM and BiLST+att (rows 3 & 4, rows 7 & 8). However, for the BERT models emb_concat() exhibits an 84% increase over emb_prod() (row 11 vs. 12). When combining the embedded entities with ALBERT and RoBERTa, we also observe that emb_concat() consistently exhibits a performance increase over emb_prod() (row 15 vs. 16; row 19 vs. 20). When tested on the CLEF CheckThat! 2021 dataset, Table 4.12 shows that all language models are markedly enhanced when combined with entity embedding models, compared to using the respective language model alone. Moreover, emb_concat() outperforms emb_prod() on all BERT-related language models (row 7 vs. 8,;row 11 vs. 12; row 15 vs. 16; row 19 vs. 20; row 23 vs. 24). When combined with SVM(TF.IDF), emb_prod() also outperforms emb_concat(). Thus, we conclude that for the ranking task, the emb_concat() model is more effective than emb-prob, and both embedding methods are more effective than the entity similarity baseline using similarity() (rows 2, 6, 10, 14, 18 in Table 4.11 and rows 2, 6, 10, 14, 18, 22 in Table 4.12).

Overall, in answer to RQ 4.3, we conclude that using embedding entities obtained from the KG embedding models, regardless of the representation method, improves all three BERT-based language representations better than the entity similarity information using similarity(), with emb_concat() exhibiting the highest effectiveness on both for the classification task (using the CheckThat! 2019 and 2021 dataset) and the ranking task (using the CheckThat! 2019, 2020, & 2021 datasets).

4.4.4 RQ 4.4: KG Embedding Model

RQ 4.4 identifies the most effective KG embedding model $KG()$ among Wikipedia2Vec, TransE, TransR, RESCAL, DISTMult, and ComplEx in assisting the language models to identify check-worthy claims. This research question addresses **Limitation G3**, which concerns finding the most suitable entity embedding model for the check-worthy sentences and tweets task. To address RQ 4.4, Tables 4.13 and 4.14 present the classification results obtained on the CLEF CheckThat! 2019 dataset and on the 2021 tweets dataset, respectively; while Tables 4.15 and 4.16 present results on the ranking task, on the CLEF CheckThat! 2019 dataset and on the 2021 tweet dataset, respectively.

Table 4.13 shows the results obtained by combining different KG entity embedding models with the various language representations for the classification task on the 2019 dataset. We observe that ComplEx does not significantly outperform Wikipedia2Vec when combined with SVM(TF.IDF) (row 6 vs. 1), but consistently and significantly outperforms

Table 4.13: Classification performances on the CheckThat! 2019 dataset, using emb_concat() as entity representation combination method, while alternating language models $LM()$ and entity embedding models $KG()$. **Bold** indicates the best performance; Numbers in the Significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	$LM()$	$KG()$	P	R	F1	Significance
1	SVM(TF.IDF)	Wikipedia2Vec	0.06	0.05	0.05	-
2	SVM(TF.IDF)	TransE	0.06	0.05	0.05	-
3	SVM(TF.IDF)	TransR	0.06	0.05	0.05	-
4	SVM(TF.IDF)	RESCAL	0.06	0.05	0.05	-
5	SVM(TF.IDF)	DistMult	0.07	0.05	0.06	-
6	SVM(TF.IDF)	ComplEx	0.07	0.05	0.06	-
7	BiLSTM+att	Wikipedia2Vec	0.13	0.10	0.11	1-6
8	BiLSTM+att	TransE	0.11	0.08	0.09	1-6
9	BiLSTM+att	TransR	0.12	0.08	0.09	1-6
10	BiLSTM+att	RESCAL	0.12	0.08	0.10	1-6,8,9
11	BiLSTM+att	DistMult	0.13	0.12	0.12	1-10
12	BiLSTM+att	ComplEx	0.14	0.13	0.13	1-10
13	BERT	Wikipedia2Vec	0.19	0.11	0.14	1-11
14	BERT	TransE	0.19	0.10	0.13	1-11
15	BERT	TransR	0.19	0.11	0.14	1-11
16	BERT	RESCAL	0.19	0.11	0.14	1-11
17	BERT	DistMult	0.19	0.12	0.15	1-16
18	BERT	ComplEx	0.20	0.13	0.15	1-16
19	ALBERT	Wikipedia2Vec	0.22	0.15	0.18	1-18
20	ALBERT	TransE	0.22	0.14	0.17	1-18
21	ALBERT	TransR	0.23	0.14	0.18	1-18
22	ALBERT	RESCAL	0.24	0.15	0.19	1-21, 25-28
23	ALBERT	DistMult	0.24	0.15	0.19	1-21, 25-28
24	ALBERT	ComplEx	**0.25**	**0.16**	**0.20**	1-22, 25-30
25	RoBERTa	Wikipedia2Vec	0.21	0.15	0.17	1-18
26	RoBERTa	TransE	0.21	0.14	0.16	1-18
27	RoBERTa	TransR	0.21	0.15	0.17	1-18
28	RoBERTa	RESCAL	0.20	0.14	0.16	1-18
29	RoBERTa	DistMult	0.23	0.15	0.18	1-18, 25-28
30	RoBERTa	ComplEx	0.24	0.14	0.18	1-18 25-28

Wikipedia2Vec, TransE, TransR and RESCAL (row 12 vs. rows 7-10; row 18 vs. 13-16; row 24 vs. 19-21; row 30 vs. 25-28) for all the neural language representation models we use. However, while ComplEx does not significantly outperform DistMult, across all language representation models, it does exhibit an average of 1% absolute improvement in F1 over the DistMul KG embeddings (see row 12 vs. 11, row 18 vs. 17, row 24 vs. 23, row 30 vs. 29). The results are expected, given previous reported results in the literature [180, 195], since ComplEx and DistMult indeed outperform other KG embedding models on the link prediction task. Similarly, Table 4.14 shows the results of a combination of the KG embedding models with the language models, for the classification task on the 2021 tweets dataset.

Table 4.14: Classification performances on the CheckThat! 2021 tweets dataset, using emb_concat() as entity representation combination method, while alternating language models $LM()$ and entity embedding models $KG()$. **Bold** indicates the best performance; Numbers in the Significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	$LM()$	$KG()$	P	R	F1	Significance
1	SVM(TF.IDF)	Wikipedia2Vec	0.08	0.16	0.10	-
2	SVM(TF.IDF)	TransE	0.08	0.16	0.10	-
3	SVM(TF.IDF)	TransR	0.07	0.16	0.11	-
4	SVM(TF.IDF)	RESCAL	0.08	0.16	0.11	-
5	SVM(TF.IDF)	DistMult	0.08	0.16	0.11	-
6	SVM(TF.IDF)	ComplEx	0.08	0.16	0.11	-
7	BiLSTM+att	Wikipedia2Vec	0.10	0.21	0.13	1-6
8	BiLSTM+att	TransE	0.10	0.21	0.14	1-6
9	BiLSTM+att	TransR	0.10	0.21	0.14	1-6
10	BiLSTM+att	RESCAL	0.11	0.21	0.14	1-6
11	BiLSTM+att	DistMult	0.11	0.21	0.14	1-6
12	BiLSTM+att	ComplEx	0.12	0.26	0.17	1-11, 13,19,25
13	BERT	Wikipedia2Vec	0.10	0.21	0.13	1-6
14	BERT	TransE	0.10	0.21	0.14	1-6
15	BERT	TransR	0.11	0.21	0.14	1-6
16	BERT	RESCAL	0.12	0.26	0.17	1-11,13-15, 19,25
17	BERT	DistMult	0.13	0.26	0.17	1-11,13-15, 19,25
18	BERT	ComplEx	0.13	0.32	0.18	1-11,13-17, 19,25
19	ALBERT	Wikipedia2Vec	0.11	0.21	0.14	1-6
20	ALBERT	TransE	0.12	0.26	0.17	1-11,13-15, 19,25
21	ALBERT	TransR	0.13	0.26	0.17	1-11,13-15, 19,25
22	ALBERT	RESCAL	0.13	0.32	0.18	1-11,13-17, 19-21,25
23	ALBERT	DistMult	0.13	0.32	0.18	1-11,13-17, 19-21,25
24	ALBERT	ComplEx	0.14	0.32	0.19	1-23
25	RoBERTa	Wikipedia2Vec	0.11	0.21	0.14	1-6
26	RoBERTa	TransE	0.12	0.26	0.17	1-11,13-15, 19,25
27	RoBERTa	TransR	0.12	0.26	0.17	1-11,13-15, 19,25
28	RoBERTa	RESCAL	0.12	0.32	0.17	1-11,13-15, 19,25
29	RoBERTa	DistMult	0.13	0.32	0.18	1-11,13-17, 19-21,25
30	RoBERTa	ComplEx	0.13	0.32	0.18	1-11,13-17, 19-21,25
31	BERTweet	Wikipedia2Vec	0.18	0.58	0.27	1-30
32	BERTweet	TransE	0.17	0.58	0.26	1-30
33	BERTweet	TransR	0.18	0.58	0.27	1-30
34	BERTweet	RESCAL	0.17	0.63	0.27	1-33
35	BERTweet	DistMult	**0.18**	**0.63**	**0.28**	1-33
36	BERTweet	ComplEx	**0.18**	**0.63**	**0.28**	1-33

We observe that when combined with SVM(TF.IDF), different KG embedding models do not perform significantly (McNemar's Test, $p<0.01$) differently from each other (rows 1-6). When combined with BERTweet, DistMult and ComplEx outperform Wikipedia2Vec, TransE, TransR and RESCAL (rows 35 & 36 vs. rows 31-34), and obtain the equal best

Table 4.15: Ranking performances on the CheckThat! 2019 dataset, using emb_concat() as entity representation combination method, while alternating language models $LM()$ and entity embedding models $KG()$. **Bold** indicates the best performance.

#	$LM()$	$KG()$	MAP	MRR	P@1	P@5	P@10	P@20	P@50
1	SVM(TF.IDF)	Wikepedia2Vec	0.1332	0.3361	0.3254	0.2000	0.2000	0.1286	0.0915
2	SVM(TF.IDF)	TransE	0.1332	0.3361	0.3254	0.2000	0.2000	0.1286	0.0915
3	SVM(TF.IDF)	TransR	0.1263	0.5714	0.2857	0.1714	0.1429	0.0929	0.0929
4	SVM(TF.IDF)	RESCAL	0.1453	0.4176	**0.3361**	0.2857	0.2571	0.2000	**0.2286**
5	SVM(TF.IDF)	DISTMult	0.1453	0.3158	0.2857	0.2857	0.2000	0.2286	0.2000
6	SVM(TF.IDF)	ComplEx	0.1496	0.4187	0.3098	0.2857	0.2571	0.2000	0.1286
7	BiLSTM+att	Wikepedia2Vec	0.0659	0.3361	0.2857	0.1429	0.1429	0.0714	0.0314
8	BiLSTM+att	TransE	0.0659	0.3158	0.2857	0.1429	0.1429	0.1429	0.0714
9	BiLSTM+att	TransR	0.0715	0.3158	0.2432	0.1429	0.1286	0.0714	0.0314
10	BiLSTM+att	RESCAL	0.0659	0.3361	0.2857	0.1429	0.1429	0.0714	0.0314
11	BiLSTM+att	DISTMult	0.0659	0.3158	0.2000	0.1429	0.1429	0.1286	0.0714
12	BiLSTM+att	ComplEx	0.0715	0.2257	0.1286	0.1429	0.1429	0.1857	0.1343
13	BERT	Wikepedia2Vec	0.1011	**0.6196**	**0.3361**	0.1714	0.1429	0.0929	0.0686
14	BERT	TransE	0.1011	0.5714	0.3098	0.2000	0.1714	0.1286	0.0929
15	BERT	TransR	0.1011	**0.6196**	0.3098	0.1714	0.0929	0.0929	0.0686
16	BERT	RESCAL	0.1263	0.5714	0.2857	0.1714	0.1429	0.0929	0.0929
17	BERT	DISTMult	0.1263	**0.6196**	0.3098	0.2571	0.1429	0.0929	0.0929
18	BERT	ComplEx	0.1453	**0.6196**	**0.3361**	0.2857	0.1714	0.1286	0.0929
19	ALBERT	Wikepedia2Vec	0.1580	**0.6196**	0.3098	0.2857	0.2571	0.2286	**0.2286**
20	ALBERT	TransE	0.1332	0.4176	**0.3361**	0.1429	0.1429	0.1286	0.0929
21	ALBERT	TransR	0.1263	0.3158	0.3098	0.2000	0.2286	0.1286	0.0929
22	ALBERT	RESCAL	0.1332	0.5714	0.3098	0.2286	0.2000	0.1286	0.0929
23	ALBERT	DISTMult	0.1580	0.4176	0.2857	0.2000	0.1429	0.1429	0.0929
24	ALBERT	ComplEx	**0.1821**	**0.6196**	**0.3361**	**0.3098**	**0.2857**	**0.2571**	0.1286
25	RoBERTa	Wikepedia2Vec	0.1453	0.4176	**0.3361**	0.2857	0.2571	0.2000	0.2286
26	RoBERTa	TransE	0.1332	0.4176	0.2857	0.2571	0.2000	0.2000	0.1286
27	RoBERTa	TransR	0.1263	0.4176	0.2000	0.2857	0.2000	0.2286	0.2000
28	RoBERTa	RESCAL	0.1453	0.3158	0.2857	0.2857	0.2000	0.2286	0.2000
29	RoBERTa	DISTMult	0.1496	0.4187	0.3098	0.2857	0.2571	0.2000	0.1286
30	RoBERTa	ComplEx	0.1660	0.5714	**0.3361**	**0.3098**	0.2000	**0.2571**	**0.2286**

performances among all the language models and KG embedding combinations. For the BiLSTM+att, BERT, ALBERT, and RoBERTa language models, the ComplEx model significantly outperforms all other KG entity embeddings (rows 7-11 vs. 12; rows 13-17 vs. 18; rows 19-23 vs. 24; and rows 25-29 vs. 30).

For the ranking task, Table 4.15 shows that ALBERT + ComplEx achieves the best performance among our experiments, obtaining 0.1821, a tie with the best performing run in the official leaderboard, on the 2019 dataset. Moreover, it also shows that ALBERT + ComplEx obtains the highest MAP in the 2020 dataset among all the models we tested, as well as the models in the leaderboard[34]. Under further investigation, we found that ALBERT + ComplEx successfully identified the single check-worthy sentence within one debate of the test set, and therefore obtained the highest improvement on MAP. For the ranking task on the 2021 dataset, Table 4.16 shows that BERTweet + ComplEx achieves the best performance among our experiments, obtaining 0.3681 (row 36), a remarkable improvement over

Table 4.16: Ranking performances on the CheckThat! 2021 Tweets dataset, using emb_concat() as entity representation combination method, while alternating language models $LM()$ and entity embedding models $KG()$. **Bold** indicates the best performance.

#	$LM()$	$KG()$	MAP	MRR	P@1	P@5	P@10	P@20	P@50
1	SVM(TF.IDF)	Wikepedia2Vec	0.0635	0.1429	0.0000	0.0100	0.1000	0.0500	0.0400
2	SVM(TF.IDF)	TransE	0.1147	0.5000	0.0000	0.2000	0.3000	0.1000	0.0600
3	SVM(TF.IDF)	TransR	0.1203	0.5000	0.0000	0.2000	0.3000	0.1500	0.0600
4	SVM(TF.IDF)	RESCAL	0.1205	0.5000	0.0000	0.2000	0.3000	0.1500	0.0600
5	SVM(TF.IDF)	DISTMult	0.1295	0.5000	0.0000	0.4000	0.3000	0.1500	0.0600
6	SVM(TF.IDF)	ComplEx	0.1824	**1.0000**	**1.0000**	0.4000	0.3000	0.1500	0.0600
7	BiLSTM+att	Wikepedia2Vec	0.1149	0.5000	0.0000	0.2000	0.2000	0.1500	0.0800
8	BiLSTM+att	TransE	0.1215	0.3333	0.0000	0.2000	0.3000	0.1500	0.1000
9	BiLSTM+att	TransR	0.1303	0.5000	0.0000	0.2000	0.3000	0.1500	0.100
10	BiLSTM+att	RESCAL	0.1391	0.5000	0.0000	0.4000	0.3000	0.1500	0.1000
11	BiLSTM+att	DISTMult	0.1654	**1.0000**	**1.0000**	0.4000	0.3000	0.1500	0.1000
12	BiLSTM+att	ComplEx	0.2015	0.5000	0.0000	0.6000	**0.5000**	0.2500	0.1000
13	BERT	Wikepedia2Vec	0.1813	**1.0000**	**1.0000**	0.6000	0.3000	0.1500	0.0800
14	BERT	TransE	0.2030	**1.0000**	**1.0000**	0.4000	0.3000	0.2000	0.1000
15	BERT	TransR	0.2034	**1.0000**	**1.0000**	0.4000	0.3000	0.2000	0.1000
16	BERT	RESCAL	0.2197	**1.0000**	**1.0000**	0.4000	0.4000	0.2500	0.1000
17	BERT	DISTMult	0.2550	**1.0000**	**1.0000**	0.6000	0.4000	0.2500	0.1000
18	BERT	ComplEx	0.2614	**1.0000**	**1.0000**	0.6000	0.4000	0.2500	0.1000
19	ALBERT	Wikepedia2Vec	0.1823	**1.0000**	**1.0000**	0.6000	0.3000	0.1500	0.0800
20	ALBERT	TransE	0.2486	**1.0000**	**1.0000**	0.6000	0.4000	0.2000	0.1000
21	ALBERT	TransR	0.2493	**1.0000**	**1.0000**	0.6000	0.4000	0.2000	0.1000
22	ALBERT	RESCAL	0.2836	**1.0000**	**1.0000**	0.6000	0.4000	0.3000	0.1200
23	ALBERT	DISTMult	0.2839	**1.0000**	**1.0000**	0.6000	0.4000	0.3000	0.1200
24	ALBERT	ComplEx	0.2887	**1.0000**	**1.0000**	0.6000	**0.5000**	0.3000	0.1200
25	RoBERTa	Wikepedia2Vec	0.2182	**1.0000**	**1.0000**	0.0600	0.4000	0.2000	0.0800
26	RoBERTa	TransE	0.2675	**1.0000**	**1.0000**	0.6000	0.4000	0.2500	0.1000
27	RoBERTa	TransR	0.2680	**1.0000**	**1.0000**	0.6000	0.4000	0.2500	0.1000
28	RoBERTa	RESCAL	0.2766	**1.0000**	**1.0000**	0.6000	0.4000	0.3000	0.1200
29	RoBERTa	DISTMult	0.2742	**1.0000**	**1.0000**	0.0600	0.4000	0.2000	0.1000
30	RoBERTa	ComplEx	0.2787	**1.0000**	**1.0000**	0.6000	0.4000	0.3000	0.1200
31	BERTweet	Wikepedia2Vec	0.3268	**1.0000**	**1.0000**	**0.8000**	0.4000	**0.3500**	0.2200
32	BERTweet	TransE	0.3660	**1.0000**	**1.0000**	0.6000	**0.5000**	**0.3500**	0.2200
33	BERTweet	TransR	0.3584	**1.0000**	**1.0000**	0.6000	0.4000	**0.3500**	0.2200
34	BERTweet	RESCAL	0.3578	**1.0000**	**1.0000**	0.4000	0.4000	0.2500	0.1000
35	BERTweet	DISTMult	0.3607	**1.0000**	**1.0000**	0.6000	0.4000	**0.3500**	0.2200
36	BERTweet	ComplEx	**0.3681**	**1.0000**	**1.0000**	0.6000	0.4000	**0.3500**	**0.2400**

the best performing run in the official leaderboard (row 32 in Table 4.8). This result suggests that entities embeddings obtained from the ComplEx model can indeed help language model identify the most check-worthy tweets. We further conducted a case study in order to understand why ComplEx can achieve the best performance consistently. Table 4.17 presents 2 cases where the ComplEx model togehter with the ALBERT language model successfully identified the check-worthy sentences, while other entity embedding models did not. Upon further investigation, we found that in these two cases, the two entities presented in the sentences are not directly related to each other. We therefore postulate that ComplEx is able to identify hidden relations between two weakly associated entities better than other entity

embedding models.

Overall, we conclude that among all 6 KG embedding models we tested, ComplEx produces consistently the highest performance.

Table 4.17: Two sentences that are correctly identified as check-worthy using ALBERT, ComplEx entity embedding model, and emb_concat() model, but are otherwise not identified.

Speaker	Sentence	Entity 1	Entity 2
Donald Trump	They want to take away your good health care, and essentially use socialism to turn America into Venezuela and Democrats want to totally open the borders.	Venezuela	Democrat
Donald Trump	And one state said – you know, it was interesting, one of the states we won, Wisconsin – I didn't even realize this until fairly recently – that was the one state Ronald Reagan didn't win when he ran the board his second time.	Wisconsin	Ronald Reagan

4.4.5 Failure Analysis

In this section, we aim to identify the bottleneck of our model, on the task of check-worth sentences identification.

Table 4.18 shows that there are different numbers of transcripts and check-worthy sentences from different parties that participated in the debate. That is, check-worthy sentences from interviews and speeches given by Trump make up as much as 60% of the total number of check-worthy sentences. Moreover, we observe sentences from Democrat candidates are classified more accurately, than sentences from Republic candidates. For example, our best performing classification (ALBERT + ComplEx) achieves 0.31 on the Democratic debates, which have a much higher number of entities detected per check-worth sentences (2.6), compared to the 0.15 recall on Republican candidates, which have only 2 entities per check-worthy sentence. Furthermore, transcripts considering Trump alone has only 1.16 entities on average, with a recall of 0.11.

To illustrate where our system fails, Table 4.19 shows 6 sentences, with the number of identified entities from each sentence, and if the ALBERT + ComplEx classifier correctly identified the sentence as check-worthy. We observe that in three of false negative cases (row

Table 4.18: Descriptive analysis of the test set of 2020 dataset. Note, this table consist of only check-worthy sentences (denoted as CW). The results we investigate here is obtained using ALBERT language model, ComplEx entity embedding method, and emb_comcat method.

debate type	# of transcript	# of cw	cw/transcript	# of entities/CW	Recall (classification)
Democratic	4	26	6.5	2.62	0.31
Republican	1	7	7	2	0.15
Mixed	2	23	11.5	2.57	0.17
Trump alone	13	83	6.38	1.16	0.11

Table 4.19: A selected cases of check-worthy sentences, the identified entities, and if ALBERT + ComplEx successfully identified it as check-worthy. **Bold** denotes the identified entities.

Speaker	Sentence	# of entities	Predicted correctly
Trump	**Trump** was totally against the war in **Iraq**.	2	Y
Trump	But when you make your car or when you make your **air conditioner**, and you think you're going to fire all of our workers and open up a new place in another country, and you're going to come through what will be a very strong border, which is already – you see what's happened; 61 percent down now in terms of illegal people coming in.	1	N
Cruz	**Bernie** helped write **Obamacare**.	2	Y
Cruz	There are many people in **America** struggling with exactly what you are, in the wreckage of **Obamacare**, with skyrocketing premiums, with deductibles that are unaffordable, and with really limited care.	2	N
Clinton	**Trump**'s on record extensively supporting intervention in **Libya**, when **Gadhafi** was threatening to massacre his population.	3	Y
Clinton	And I do think there is an agenda out there, supported by my opponent, to do just that.	0	N

2,4, and 6), there are only less than or equal to 2 entities, whereas the correctly identified check-worthy sentences have more than 2 entities. Therefore, we postulate that the number of entities per sentence can indeed affect the performance of our proposed model.

4.4.6 Recap of Main Findings

In this section, we recap our main findings for RQs 4.1-4.4 and indicate the implications of our study. Tables 4.20 and 4.21 summarise the performance of a salient subset of approaches on the classification and ranking tasks, respectively.

For each language model, the summarising tables present results obtained using three conditions: language model only; with entity pair representation using the Wikipedia2Vec KG embedding model and using the emb_concat() method; and with entity pair representation using the ComplEx embedding and using the emb_concat() method. We do not include the emb_prod() method in our summarising tables, as our results for RQ 4.3 showed that emb_concat() consistently outperforms emb_prod() across the CheckThat! 2019 and 2020 datasets on both the classification and ranking tasks (see Section 4.4.3).

For the classification task (Table 4.20, on the 2019 and 2021 datasets), we highlight our conclusion from RQ 4.1 (see Section 4.4.1) that the ALBERT language model (rows 11 - 13) significantly outperforms all other language models for check-worthy sentences classification. For the check-worthy tweets classification task, BERTweet performs the best among all

Table 4.20: Summary of classification performances on the CheckThat! 2019 and 2021 datasets. **Bold** indicates the best performance; Numbers in the column *Significance* indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	$LM()$	$KG()$	$COM()$	P	R	F1	Significance
CLEF'2019 CheckThat! results							
1	Random Classifier	-	-	0.01	0.01	0.01	-
2	SVM(TF.IDF)	-	-	0.01	0.01	0.01	-
3	SVM(TF.IDF)	Wikipedia2Vec	emb_concat()	0.06	0.05	0.05	1,2
4	SVM(TF.IDF)	ComplEx	emb_concat()	0.07	0.05	0.06	1-2
5	BiLSTM+att	-	-	0.12	0.07	0.09	1-4
6	BiLSTM+att	Wikipedia2Vec	emb_concat()	0.13	0.10	0.11	1-5
7	BiLSTM+att	ComplEx	emb_concat()	0.14	0.13	0.13	1-6
8	BERT	-	-	0.12	0.09	0.10	1-5
9	BERT	Wikipedia2Vec	emb_concat()	0.19	0.11	0.14	1-8
10	BERT	ComplEx	emb_concat()	0.20	0.13	0.15	1-9
11	ALBERT	-	-	0.14	0.11	0.12	1-6,8
12	ALBERT	Wikipedia2Vec	emb_concat()	0.22	0.15	0.18	1-10,13
13	ALBERT	ComplEx	emb_concat()	**0.25**	**0.16**	**0.20**	1-12,14-16
14	RoBERTa	-	-	0.14	0.11	0.11	1-6,8
15	RoBERTa	Wikipedia2Vec	emb_concat()	0.21	0.15	0.17	1-11,14
16	RoBERTa	ComplEx	emb_concat()	0.24	0.14	0.18	1-12,14,15
CLEF'2021 CheckThat! results							
17	Random Classifier	-	-	0.05	0.05	0.05	-
18	SVM(TF.IDF)	-	-	0.05	0.11	0.07	-
19	SVM(TF.IDF)	Wikipedia2Vec	emb_concat()	0.08	0.16	0.105	1,2
20	SVM(TF.IDF)	ComplEx	emb_concat()	0.08	0.16	0.11	1-2
21	BiLSTM+att	-	-	0.05	0.11	0.07	1-4
22	BiLSTM+att	Wikipedia2Vec	emb_concat()	0.10	0.21	0.14	1-5
23	BiLSTM+att	ComplEx	emb_concat()	0.12	0.26	0.17	1-6
24	BERT	-	-	0.08	0.16	0.10	1-5
25	BERT	Wikipedia2Vec	emb_concat()	0.11	0.21	0.14	1-8
26	BERT	ComplEx	emb_concat()	0.13	0.32	0.18	1-9
27	ALBERT	-	-	0.08	0.16	0.11	1-6,8
28	ALBERT	Wikipedia2Vec	emb_concat()	0.11	0.21	0.148	1-10,13
29	ALBERT	ComplEx	emb_concat()	0.14	0.32	0.19	1-12,14-16
30	RoBERTa	-	-	0.09	0.16	0.11	1-6,8
31	RoBERTa	Wikipedia2Vec	emb_concat()	0.11	0.21	0.14	1-11,14
32	RoBERTa	ComplEx	emb_concat()	0.13	0.32	0.18	1-12,14,15
33	BERTweet	-	-	0.16	0.47	0.23	1-6,8
34	BERTweet	Wikipedia2Vec	emb_concat()	0.18	0.58	0.27	1-11,14
19	BERTweet	ComplEx	emb_concat()	**0.18**	**0.63**	**0.28**	1-12,14,15

language models. We also confirm our conclusion from RQ 4.2 (see Section 4.4.2) that entity embeddings improve the language models performance at identifying check-worthy sentences (row 3 & 4 vs. 2, rows 6 & 7 vs. 5, rows 9 & 10 vs. 8, rows 12 & 13 vs. 11, rows 15 & 16 vs. 14). Finally, we reiterate our conclusion from RQ 4.4 (see Section 4.4.4) that the ComplEx embedding method (rows 4, 7, 10, 13 & 16) – which uses the facts-alone KG embedding

Table 4.21: Summary of ranking performance on CLEF' 2019, 2021, & 2020 CheckThat! dataset. **Bold** denotes the best performance for a given measure in a given year.

#	$LM()$	$KG()$	$COM()$	MAP	MRR	P@1	P@5	P@10	P@20	P@50
Experimental Results using CLEF'2019 CheckThat! dataset										
1	SVM(TF.IDF)	-	-	0.1193	0.3513	0.1429	0.2571	0.1571	0.1714	0.1086
2	SVM(TF.IDF)	Wikepedia2Vec	emb_concat()	0.1332	0.3361	0.3254	0.2000	0.2000	0.1286	0.0915
3	SVM(TF.IDF)	ComplEx	emb_prod()	0.1332	0.3158	0.3098	0.2000	0.2571	0.1429	0.0929
4	BiLSTM+att	-	-	0.1455	0.2432	0.1429	0.1429	0.1429	0.1857	0.1343
5	BiLSTM+att	Wikepedia2Vec	emb_concat()	0.0659	0.3361	0.2857	0.1429	0.1429	0.0714	0.0314
6	BiLSTM+att	ComplEx	emb_concat()	0.0715	0.2257	0.1286	0.1429	0.1429	0.1857	0.1343
7	BERT	-	-	0.0715	0.2257	0.1429	0.2000	0.1286	0.0857	0.0600
8	BERT	Wikepedia2Vec	emb_concat()	0.1011	**0.6196**	**0.3361**	0.1714	0.1429	0.0929	0.0686
9	BERT	ComplEx	emb_concat()	0.1011	**0.6196**	**0.3361**	0.2857	0.1714	0.1286	0.0929
10	ALBERT	-	-	0.1332	0.4176	0.3098	0.2000	0.1429	0.1286	0.0929
11	ALBERT	Wikepedia2Vec	emb_concat()	0.1580	**0.6196**	0.3098	0.2857	0.2571	**0.2286**	**0.2286**
12	ALBERT	ComplEx	emb_concat()	**0.1821**	**0.6196**	**0.3361**	**0.3098**	**0.2857**	0.2571	0.0929
13	RoBERTa	-	-	0.1011	0.3158	0.2286	0.2000	0.1429	0.1286	0.0929
14	RoBERTa	Wikepedia2Vec	emb_concat()	0.1453	0.4176	**0.3361**	0.2857	0.2571	0.2000	**0.2286**
15	RoBERTa	ComplEx	emb_concat()	0.1660	0.5174	**0.3361**	**0.3098**	0.2000	0.2571	**0.2286**
Experimental results using CLEF'2020 CheckThat! dataset										
16	SVM(TF.IDF)	-	-	0.0946	0.1531	0.0000	0.0600	0.0400	0.0450	0.0240
17	SVM(TF.IDF)	ComplEx	emb_concat()	0.0923	0.1170	0.0000	0.0200	0.0500	0.0675	0.0270
18	BiLSTM+att	-	-	0.0151	0.0320	0.0000	0.0100	0.0150	0.0075	0.0090
19	BiLSTM+att	ComplEx	emb_concat()	0.0183	0.0320	0.0000	0.0200	0.0100	0.0100	0.0090
20	BERT	-	-	0.0262	0.0819	0.0500	0.0300	0.0250	0.0125	0.0110
21	BERT	ComplEx	emb_concat()	0.0373	0.0819	0.0500	0.0500	0.0350	0.0175	0.0130
22	ALBERT	-	-	0.0537	0.2145	0.2000	0.0800	0.0500	0.0250	0.1600
23	ALBERT	ComplEx	emb_concat()	**0.1036**	**0.2644**	**0.2500**	**0.0900**	**0.0550**	**0.0275**	**0.0170**
24	RoBERTa	-	-	0.0424	0.1315	0.1000	0.6000	0.0400	0.0200	0.1400
25	RoBERTa	ComplEx	emb_concat()	0.0923	0.1814	0.1500	0.0700	0.0450	0.0225	0.0150
Experimental results using CLEF'2021 CheckThat! dataset										
26	SVM(TF.IDF)	-	-	0.0608	0.1111	0.0000	0.0000	0.1000	0.0500	0.0400
27	SVM(TF.IDF)	ComplEx	emb_concat()	0.1824	**1.0000**	**1.0000**	0.4000	0.3000	0.1500	0.0600
28	BiLSTM+att	-	-	0.0635	0.1429	0.0000	0.0100	0.1000	0.0500	0.04000
29	BiLSTM+att	ComplEx	emb_concat()	0.2015	0.5000	0.0000	0.6000	**0.5000**	0.2500	0.1000
30	BERT	-	-	0.0757	0.1429	0.0000	0.0000	0.1000	0.1000	0.0600
31	BERT	ComplEx	emb_concat()	0.2614	**1.0000**	**1.0000**	0.6000	0.4000	0.2500	0.1000
32	ALBERT	-	-	0.0777	0.1429	0.0000	0.0000	0.1000	0.1000	0.0600
33	ALBERT	ComplEx	emb_concat()	0.2887	**1.0000**	**1.0000**	0.6000	**0.5000**	0.3000	0.1200
34	RoBERTa	-	-	0.0806	0.1429	0.0000	0.0000	0.2000	0.1000	0.0600
35	RoBERTa	ComplEx	emb_concat()	0.2787	**1.0000**	**1.0000**	0.6000	0.4000	0.3000	0.1200
36	BERTweet	-	-	0.1326	0.5000	0.0000	0.2000	0.1000	0.1500	0.1800
37	BERTweet	ComplEx	emb_concat()	**0.3681**	**1.0000**	**1.0000**	0.6000	0.4000	**0.3500**	**0.2400**

– significantly outperforms the semantic KG embedding method (i.e., Wikipedia2Vec, rows 3, 6, 9, 12 & 15).

For the ranking task (Table 4.21, on 2019, 2020 & 2021 datasets), we draw similar conclusions as for the classification task: the ALBERT language model (rows 10 - 12 for the 2019 dataset, rows 22 & 23 for the 2020 dataset) consistently outperforms all other language models on ranking check-worthy sentences, and BERTweet outperforms all other language models (including ALBERT) on ranking check-worthy tweets. The ComplEx embedding model (rows 3, 6, 9, 12, 15 for the 2019 dataset, 17, 19, 21, 23, 25 for the 2020 dataset) consistently outperforms all other KG embedding models. Moreover, ALBERT + ComplEx + emb_concat() (row 11 for the 2019 dataset, row 20 for the 2020 dataset) obtains the best

performance among all tested models on ranking check-worthy sentences, while BERTweet + ComplEx + emb_concat() performs the best on ranking check-worthy tweets.

Thus, we conclude that ALBERT + ComplEx + emb_concat() can best identify and rank check-worthy sentences in a given speech or debate transcript. On the other hand, BERTweet + ComplEx + emb_concat() can best identify and rank check-worthy tweets among a sample of tweets. In short, the findings of our study can thus be summarised as follows:

- In response to RQ 4.1, we conclude that deep neural language models help identify the sentences/tweets that require further manual fact-checking;
- In response to RQ 4.2, we conclude that embedded entities within sentences/tweets help identify the sentences/tweets that require further manual fact-checking;
- In response to RQ 4.3, we conclude that the most effective way to combine an entity pair representation with a text representation is to concatenate the two vectors together;
- In response to RQ 4.4, we conclude that the tested facts-alone KG embedding models perform better than tested semantic KG embedding method (i.e., Wikipedia2Vec). The best performing KG embedding model in our study is the ComplEx model.
- Finally, failure analysis shows that the performance of our model is affected by the number of entities present in the sentence.

4.5 Conclusions

In this chapter, we proposed a uniform model for the task of check-worthy sentence/tweet identification in Section 4.2, formulated as either a classification or a ranking task. We proposed to use BERT-related pre-trained language representations, and, in a novel manner, integrated entity embeddings obtained from knowledge graphs into the classifier and ranker. Our proposed model addressed several limitations we identified in Chapter 2. Namely, the proposed model directly addressed **Gap 2**, where we aim to identify check-worthy tweets and sentences, to fact-check only a subset of all the tweets circulation online. To answer RQ 4.1 and address **Limitation L2**, Section 4.4.1 concluded that in the specific task of identifying check-worthy sentences/tweets, the most effective language model among our tested models is ALBERT for sentence embedding and BERTweet for tweet embedding. To answer RQ 4.2 and RQ 4.3, and address the **Limitation L1 & G2**, in Sections 4.4.2 and 4.4.3 we concluded that we have identified that entity embeddings can improve both classification and ranking tasks, and the effective way to represent a pair of entities in a text is to concatenate two entity embeddings together. To answer RQ 4.4. and address **Limitation G3**, in Section 4.4.4 we

concluded that the KG embedding model ComplEx is the most effective entity embedding model in the task of identifying check-worthy sentences/tweets.

In our thesis statement in Section 1.3, we hypothesised that analysing embedded entities within sentences/claims or tweets can help language models to identify the check-worthy ones more accurately, from tweet content, articles, and debate quotes. In this chapter, we conclude that our model, which combines deep learning language models with embedded entity representations in a novel manner, can achieve better performances in identifying check-worthy sentences than using language models and handcrafted features alone (such as syntactic dependence [69] and Standard Universal Sentence Encoder [52]). We note that the aim of Task 1 is to identify check-worthy claims for further fact-checking, where only the identified check-worthy sentences will be further fact-checked. Thus, higher P@K metrics in the ranking task and higher Recall in the classification task is especially important. We also note that our proposed model achieves better performances on sentences/claims/tweets that contains more than 1 entities, where cautions should be paid to sentences without entities, when using our proposed entity assisted check-worthy detection model. Our extensive experiments using three public datasets from the CLEF CheckThat! 2019, 2020 and 2021 Labs demonstrate that our proposed model – based on pre-trained language models – yields state-of-the-art performances (especially in P@K metrics and Recall) in automatically identifying check-worthy sentences in political debates and speech transcripts, and check-worthy tweets spreading or commenting on the current news.

In the next chapter, we focus on task 2 in our Tweets Fact-Checking (TFC) phase. Specifically, we aim to match check-worthy claims and news titles with existing fake news datasets, in order to spot the easy to identify and recurring fake news, that are labelled as check-worthy by the WCTR phase.

Chapter 5

Assisting Fake News Detection using an Existing Fake News Collection

5.1 Introduction

In the last chapter we presented experiments in relation to Phase 1 Task 1 of our proposed framework FNDF, namely, how to identify check-worthy tweets and claims. With check-worthy tweets and sentences identified, we enter the second phase of our framework: the fact-checking phase. In this phase, we propose two tasks focusing on fake news detection, where the first task (Phase 2 Task 2) focuses on analysing the semantics of fake news and aims to detect recurrent fake news, while the second task (Phase 2 Task 3) focuses on identifying fake news using Twitter user network features. This chapter focuses on Phase 2 Task 2 of our proposed framework FNDF, and aims to tackle the task of effectively identifying recurrent fake news.

Recall the thesis statement introduced in Section 1.3, where we hypothesised that we can identify fake news by comparing a targeted claim/tweet with a set of previously debunked fake news. In Section 2.4.2, we argued that some fake news can reappear on social media and news platforms after being debunked [150, 159], and presented research on using existing fake news collection to identify recurring fake news. We also identified **Gap 3** in Section 2.4.4, which states that identifying fake news using existing fake news collections is an important but understudied task for effectively detecting fake news circulating online.

The WSDM Cup 2019 Fake News Challenge aimed to address the task of identifying recurrent fake news. This challenge required researchers to develop models that are able to predict if a given news title is agreeing with a previously debunked rumour. Inspired by this challenge, we aim to identify recurrent fake news, by comparing tweets and news articles with known fake news from existing fake news collections, to effectively identify al-

ready debunked fake news. We note that this task is similar to natural language inference (NLI) [18, 111] in natural language processing (NLP), where sentences are predicted to be logically related or not. Similarly, we also draw parallels to the task of learning semantic matching between queries and documents, where research [39, 197] have shown that a classical BM25 document weighting model can improve the performance of using semantic approaches alone.

Therefore, in this chapter, we argue that **Gap 3** can be addressed by building upon recent advances in language models for text processing, and combining the NLI task with semantic matching models. Specifically, this chapter addresses the detailed **Limitations L1 & L2** identified in Section 2.6.4. **Limitation L1** states the need to identify the most suitable language model in identifying recurring fake news. We address this limitation by comparing a range of language models in the task of identifying recurring fake news. Our experiments show that the BERT language model outperforms any other language model. **Limitation L2** states the need to combine language models with other types of features for better performances in individual tasks, such as the identification of recurring fake news. We address this limitation by combining a range of language models with the BM25 model, to build an ensemble model that account for both the semantic meanings of the texts and the semantic matching scores between the two texts. This is because the BM25 document weighting model can improve the performance of using semantic approaches alone in the task of semantic matching task [39, 197]. Our experiments show that an ensemble model of the BM25 score and the language representations indeed improve the performance of using a language model alone, on using existing fake news datasets in detecting recurring fake news.

We test our proposed ensemble mode (BM25 + language model) on two datasets – the WSDM 2019 Cup Fake News Challenge Chinese dataset and the MM-COVID English dataset. Our experiments show that the BERT language model significantly outperforms BiLSTM, which in-turn significantly outperforms a simpler embedding-based representation. Furthermore, we show that a simple BM25 feature can improve the state-of-the-art BERT approach in identifying recurring fake news. Thus, in answering the hypothesis presented in the thesis statement in Section 1.3, we show that our proposed ensemble model of the BM25 scores and language representations can accurately classify if a targeted checkworthy claim is highly similar to any existing fake news, and thus identify it as a resurfaced fake claim.

The rest of the chapter is structured as follows: Section 5.2 states the task problem, along with our proposed model to address the task. We present our experimental setup in Section 5.3, and discuss the results of the experiments in Section 5.4. Finally, we provide concluding remarks in Section 5.5.

5.2 Ensemble Model for Recurring Fake News Detection

Building upon the objective of Task 2 presented in Section 3.3.1, we describe the task we aim to tackle, and introduce our proposed ensemble model in detail.

5.2.1 Recurring Fake News Detection Task (Phase 2 Task 2)

As highlighted in Section 5.1, this chapter aims to identify the relationship between a check-worthy tweet, sentence, or news title and a debunked fake news. Table 5.1 presents the notations (a subset of the notations defined in Table 3.1) we use in this chapter:

Table 5.1: Notations used in Chapter 5.

Notation	Definition
$X_{checkworthy}$	The set of check-worthy claims identified from X
x	A sentence or tweet
FN	A collection of previously identified fake news
fn	An identified fake news in the set of identified fake news FN

We define text a entails text b as text a can be inferred from text b, according to the definition proposed by Dagan et. al. [35]. Specifically, if a tweet or sentence x entails any $fn \in FN$, we identify x as a recurring fake news. Thus, we define the task as Equation (3.11) introduced in Section 3.3.2, and repeated below:

$$RecurringFN_x = \begin{cases} 1, \text{if } \exists\ fn\ \in\ FN,\ f_{entail}\ (x, fn) = agree \\ 0, otherwise \end{cases} \tag{5.1}$$

In particular, given fn, a known fake news, and x that needs to be fact-checked, a classifier $f_{entail}()$ should classify if x *agrees* (x talks about the same news as fn), *disagrees* (x refutes the news in fn), or is *unrelated*[1], to the fn, shown as follows:

$$f_{entail}(x, fn) \rightarrow \{\textit{unrelated}, \textit{agree}, \textit{disagree}\} \tag{5.2}$$

Among these, the *agree* label indicates that x contains non-factual information that is similar to fn, while the *unrelated* and *disagreement* relationships indicate that x is not a reappearing fake news that is similar to fn, and may need to be further fact-checked. Moreover, the decision of *unrelated* vs. (*agree* $\|$ *disagree*) is equivalent to identifying relevance. We build

[1]We use these three class following the WSDM 2019 Cup Fake News Challenge dataset. In essence, these three class are similar to the *contradicts*, *entail*, and *neutral*, widely used in the NLI tasks.

upon standard text similarity approaches, as well as customised classifiers, to determine if the model $RecurringFN_x()$ can make the *agree* vs. *disagree* decision more effectively than using the language models alone.

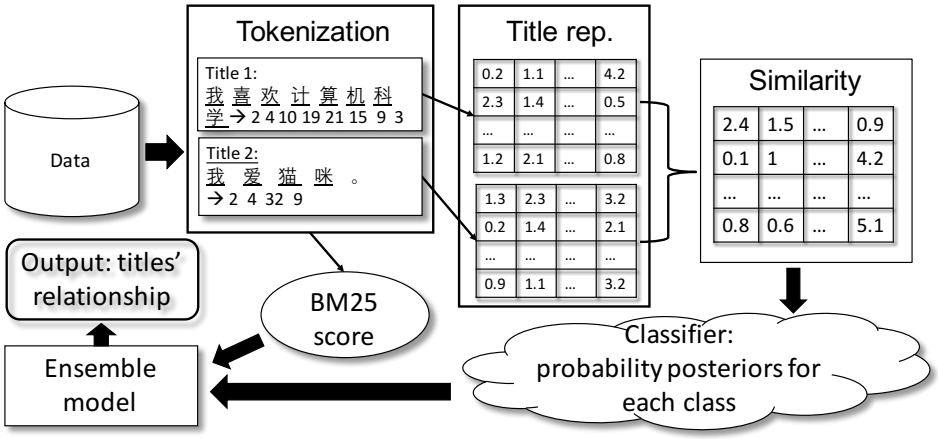


Figure 5.1: The structure and components of our model.

5.2.2 Ensemble Model for Recurring Fake News Detection

Figure 5.1 illustrates an outline of our approach, in three steps: representing terms and titles; similarity calculations; classifiers and ensembles. Note that one of the dataset this chapter uses is the WSDM Cup 2019 Fake News Challenge dataset, which is in Chinese language. We use character-level tokenisation to pre-process Chinese, because the Chinese language does not naturally have word-wise tokens. Table 5.2 shows the combinations of various used representations of terms and titles, similarity calculations and classifiers, leading to different model instantiations. The table also lists the abbreviation names given to the resulting models. For example, the BSC model uses the BERT approach to represent x and fn, the subtraction similarity (discussed in detail in Section 5.2.2.2), and the CNN classifier. Note that the ensemble models are not listed in Table 5.2, but are denoted as model abbr. + BM25. We now introduce each step separately.

5.2.2.1 Text Representation

For the WSDM 2019 Cup Fake News Challenge dataset, which is in Mandarin Chinese, we use a character-level segmentation method to transform each title into a series of tokens[2]. For the MM-COVID dataset, which is in English, we omit the segmentation step.

For all datasets, we represent each title, tweet and claim as a vector, using a range of language representation models (e.g, TF.IDF, BiLSTM, pre-trained BERT model). Thus, the textual

[2]Because we identified the character-level segmentation as the most effective in initial experiments.

Table 5.2: Models and their components used in this work.

#	Abbr.	Term/Title rep.	Similarity	Classifier
1	BL	Emb-Concat	Cosine	MLP (15 layers)
2	LR(BM25)	TF.IDF	BM25	LR
3	Emb-BiLSTM	Emb-BiLSTM	-	SoftMax layer
4	ESM	Emb-BiLSTM	Subtraction	MLP (2 layers)
5	ESC	Emb-BiLSTM	Subtraction	CNN
6	ECM	Emb-BiLSTM	Cosine	MLP (2 layers)
7	ECC	Emb-BiLSTM	Cosine	CNN
8	BERT	BERT	-	SoftMax layer
9	BSM	BERT	Subtraction	MLP (2 layers)
10	BSC	BERT	Subtraction	CNN
11	BCM	BERT	Cosine	MLP (2 layers)
12	BCC	BERT	Cosine	CNN

representations of the title, tweet and claim are obtained as such:

$$\overrightarrow{x} = LanguageModel(x) \tag{5.3}$$

$$\overrightarrow{fn} = LanguageModel(fn) \tag{5.4}$$

5.2.2.2 Text Similarity

Using the text representations $\overrightarrow{x}$ and $\overrightarrow{fn}$, we measure the similarity between x and fn using the following three methods:

1. **Cosine similarity (denoted as Cosine).** Cosine similarity measures the angle between two representations, which represents the orientation of the subjects between two text inputs [164]. The cosine similarity between x and fn is calculated as follows:

$$Sim(x, fn) = cos(\theta) = \frac{\overrightarrow{x} \cdot \overrightarrow{fn}}{|\overrightarrow{x}||\overrightarrow{fn}|} \tag{5.5}$$

2. **Vector Subtraction (denoted as Subtraction).** As mentioned in Section 2.6.2, an embedding model is able to capture the semantic information of terms. Therefore, we use a subtraction function between two titles' representations to measure the semantic distance. Note that although subtraction is not a commutative operation, it is appropriate for this task, as the relationship between x and fn is an ordered relationship[3]. The

[3]I.e., fn is existing fake news, which appeared earlier than x, thus we only classify if x *agrees* with fn, and not the other way around

subtraction between x and fn is calculated as follows:

$$Sim(x, fn) = Subtraction(x, fn) = \overrightarrow{x} - \overrightarrow{fn} \tag{5.6}$$

3. **BM25.** BM25 [146] is a weighting model that is traditionally used to score documents based on the query terms appearing in each document. In this task, we consider the claim x as a query and the existing fake news collection FN as a set of documents. We aim to find the similar fn for a given x from the existing fake news collection FN. Thus, we use BM25 to measure text similarities between x and fn, using the term frequencies rather than the vector representations of x and fn. The BM25 similarity between x and fn is calculated as follows:

$$BM25(x, fn) = \sum_{i=1}^{n} IDF\left(x_i\right) \cdot \frac{f\left(x_i, fn\right) \cdot (k_1 + 1)}{f\left(x_i, fn\right) + k_1 \cdot \left(1 - b + b \cdot \frac{|fn|}{\text{avgdl}}\right)} \tag{5.7}$$

where x_i is the i^{th} term in x, $f(x_i, fn)$ is x_i's term frequency in fn, $|fn|$ is the number of tokens in fn, and avgdl is the average token length in FN. We use the default $k_1 = 1.2$ and $b = 0.75$ in calculating the BM25. IDF(x_i) is the inverse document frequency (IDF) weight of the x term x_i, where $IDF(x_i) = ln(\frac{N - n(x_i) + 0.5}{n(x_i) + 0.5} + 1)$, N is the total number of $fn \in FN$, and $n(x_i)$ is the number of $fn \in FN$, that contains x_i.

5.2.2.3 Final Classifiers & the Ensemble Model

For the classification layer, we compare experiment results of using a dense layer with a SoftMax activation function (denoted as SoftMax), a multilayer perceptron classifier (denoted as MLP), and a convolutional neural network classifier (denoted as CNN) to classify the relationship of x and fn, and to output the predicted class. Recall Figure 5.1, where we show the outputs of the text similarities are used as inputs in the final classifier. Thus, this function is defined as follows:

$$f_{entail}(x, fn) = cls(sim(x, fn)) \tag{5.8}$$

where $cls()$ is the classification function, being either SoftMax, MLP, or CNN.

Finally, we note that integrating BM25 directly into a neural network classifier is not practical, because BM25 measures the relevance of texts at the text level and produces single scores, and the inputs to the final stage of neural networks measures the relationship of x and fn and produce vectors. Therefore, we use a logistic regression classifier (LR), which combines the BM25 score and the class posteriors of the classifier $cls()$ as input, to predict the relationship between x and fn. In particular, we choose LR because it performs the best

Table 5.3: Statistics of the WSDM 2019 Cup Fake News Challenge dataset.

Dataset	# *Unrelated*	# *Agree*	# *Disagree*	# Total
Training	198416	84626	7511	276025
Validation	8831	5406	291	14528
Testing	20897	8347	755	29999

after testing other conventional classifiers (e.g. support vector machine and Naive Bayes). Thus, the ensemble model can be defined as follows:

$$f_{entail}(x, fn) = LR(cls(sim(x, fn)), BM25(x, fn)) \tag{5.9}$$

5.3 Experimental Setup

Our experiments aim to address three research questions, namely:

- **RQ 5.1**: Which model is the most effective in learning to accurately predict the relationship between pairs of news article titles? This RQ addresses **Limitation L1**, which concerns finding the best language models for the task of identifying recurring fake news, by comparing tweets and news titles with existing fake news.
- **RQ 5.2**: Does combining the BM25 relevance score with a language model improves accuracy in predicting the relationship between pairs of news article titles? This RQ addresses **Limitation L2** presented in Section 2.6.4, which concerns enriching language models with additional information such as BM25 scores.
- **RQ 5.3**: Can our model identify recurring fake news by comparing check-worthy news with a set of debunked news? This RQ addresses **Gap 3** identified in Section 2.4.4, to validate the hypothesis that we can effectively identify recurring fake news by comparing claims, tweets, or news titles with an existing fake news collection.

5.3.1 Datasets

We use the WSDM 2019 Cup Fake News Challenge dataset[4] for RQ 5.1 and RQ 5.2, which consists of human-written Chinese news title pairs, that are labelled either *unrelated*, *agree*, or *disagree* with a given debunked fake news. All the titles are pooled from Chinese news providers or content creators. The size of the dataset, along with the number of news title and debunked fake news pairs in each class, are listed in Table 5.3.

[4] https://kaggle.com/c/fake-news-pair-classification-challenge/data

Table 5.4: Statistics of the MM-COVID dataset and the existing fake news collection.

Dataset	# Fake news	Factual news	# Total
MM-COVID Training	2056	4448	6504
MM-COVID Validation	104	218	322
MM-COVID Testing	332	645	977
Existing fake news collection	497	-	497

To answer RQ 5.3, which is identifying tweets containing recurring fake news, we use a large scale dataset called MM-COVID [96] as the experimental dataset. The MM-COVID dataset contains the source content (news articles, or original tweet making a claim) that need to be fact-checked. We construct a set of 7803 check-worthy tweets/claims $X_{checkworthy}$, consist of tweet contents and news title from the MM-COVID source content. We randomly assign 80% of the extracted source content from MM-COVID as training set, 15% as test set, and 5% as validation set. We also construct an existing fake news collection using the CoAID dataset [32]. Specifically, the CoAID dataset contains claims, news articles, and engaged tweets, that are labelled as fake and not fake. We collect the all the claims, news article titles, and engaged tweets that are labelled as fake to build the existing fake news collection FN. The constructed existing fake news collection FN contains 497 debunked fake news. Table 5.4 lists the statistics of the MM-COVID dataset and the existing fake news collection.

5.3.2 Tokenisation Method for Chinese Language

We use the WordPiece segmenter (implemented in BERT[5]) to segment each Chinese title into characters. Note that any English words in the titles remain as words. We remove the stopwords before tokenisation. We trim each title to be exactly 45 words/characters, in order to enhance the BiLSTM performance (only 11 titles in the training set exceed this length).

5.3.3 Embedding Models

BiLSTM. We use the Keras[6] implementation of bidirectional LSTMs. Each BiLSTM model has 2 layers, with 64 hidden units per layer, and a dropout rate of 0.01. We apply a Siamese style [122] embedding layer in BiLSTM approaches for x and fn, where each token is embedded into 128 dimensions, and x and fn share the same embedding layer.

BERT model. We apply the BERT-base Chinese model (12-layer, 768-hidden, 12-heads, 110M parameters) on the WSDM 2019 Cup Fake News Challenge dataset, and BERT-base-uncased English model (12-layer, 768-hidden, 12-heads, 110M parameters) on the MM-COVID dataset. Following standard practice [40], we fine-tune the BERT model on the

[5] https://github.com/google-research/bert/blob/master/tokenization.py
[6] https://keras.io

training dataset. All other parameters (e.g., learning rate, drop out rate) remain at their recommended settings. Moreover, when integrating the output of BERT into the final classification function $cls()$, we use the first token output as the input to the classification layer. For the model denoted "BERT" in Table 5.2, we use the dense layer with a SoftMax activation function as the classification layer, similar to the standard BERT model.

5.3.4 Classifiers, Baselines, and Evaluation Metrics

Classifiers: We tune all the hyper-parameters for the classifiers on the validation set of the WSDM 2019 Cup Fake News Challenge dataset. Specifically, we use the Adam optimiser with a learning rate of 0.001, and ReLU [123] as the activation function, for both the MLP and the CNN classifiers. For MLP, we use 2 layers with 64 and 16 units in each respective layer. We use 32 filters, 3 kernels, and stride 1 for CNN. We implement our models using the MarchZoo deep text matching toolbox [46][7]. We use the Sage solver, the L2 penalty, and a C regularisation score of 10 for LR.

Baseline: We train a neural network with an embedding layer, concatenate the words' vectors initiated using Word2Vec [117, 118] in a claim/tweet x into a 2D matrix, and use Cosine similarity as the similarity function, and apply a 15 layers MLP as the final classifier. We denote this baseline as BL.

Finally, we train an LR model using only BM25 scores (denoted as LR(BM25)), to demonstrate the effectiveness of using the BM25 similarity scores only.

Evaluation Metrics: We report accuracy, balanced accuracy (BAC), precision, recall, and F1 scores as evaluation metrics. Note that as presented in Table 5.3, the WSDM 2019 Cup Fake News Challenge dataset is unbalanced, where the *agree* and *disagree* classes are more important, but are smaller in size than the *unrelated* class. Therefore, we report the BAC metrics on both the WSDM 2019 Cup Fake News Challenge dataset and the MM-COVID dataset. We also report the accuracy metrics of the *agree* and *disagree* classes on the WSDM 2019 Cup Fake News Challenge dataset, and the accuracy metrics of both identifying fake news and real news on the MM-COVID dataset.

5.4 Results and Analysis

To answer RQ 5.1 & 5.2 presented in Section 5.3, we present the results of our news title relationship classification experiments. Table 5.5 presents the classification results of each model tested on the test set of the WSDM 2019 Cup Fake News Challenge dataset. To

[7] https://github.com/NTMC-Community/MatchZoo/

Table 5.5: Classification scores for WSDM 2019 Cup Fake News Challenge dataset. **Bold** denotes the best result in the table. †† denotes that an ensemble model significantly outperforms both the corresponding RNN/BERT model as well as LR(BM25) (McNemar's test, $p < 0.01$).

Model	Acc	BAC	P	R	F1	*Agree* Acc	*Disagree* Acc
BL	0.632	0.692	0.71	0.62	0.67	0.712	602
LR(BM25)	0.758	0.544	0.80	0.76	0.77	0.794	0.294
ESM	0.696	0.704	0.77	0.70	0.72	0.763	0.665
+ BM25††	0.765	0.736	0.80	0.76	0.78	0.783	0.678
ESC	0.703	0.715	0.78	0.70	0.72	0.791	0.656
+ BM25††	0.778	0.743	0.81	0.78	0.79	0.800	0.686
ECM	0.752	0.779	0.82	0.75	0.77	0.847	0.729
+ BM25††	0.762	0.782	0.83	0.76	0.78	0.879	0.804
ECC	0.789	0.758	0.83	0.79	0.80	0.809	0.687
+ BM25††	0.779	0.760	0.82	0.78	0.79	0.828	0.756
BERT	**0.885**	0.735	**0.88**	**0.89**	**0.88**	0.822	0.458
+ BM25††	0.875	0.815	**0.88**	0.87	**0.88**	0.858	0.697
BSM	0.863	0.825	**0.88**	0.86	0.87	0.877	0.736
+ BM25††	0.851	**0.847**	**0.88**	0.85	0.86	**0.892**	**0.826**
BSC	0.859	0.816	0.87	0.86	0.86	0.873	0.717
+ BM25††	0.856	0.823	0.87	0.86	0.86	0.877	0.804
BCM	0.763	0.657	0.81	0.76	0.78	0.657	0.497
+ BM25††	0.770	0.665	0.82	0.77	0.79	0.796	0.499
BCC	0.851	0.815	0.87	0.85	0.86	0.877	0.715
+ BM25††	0.845	0.826	**0.88**	0.85	0.86	0.886	0.767

answer RQ 5.3 presented in Section 5.3, which concerns whether the proposed method can identify recurring fake news, we present the results of the classification experiments on the MM-COVID dataset. Table 5.7 presents the classification results of each model tested on the test set of the MM-COVID dataset.

5.4.1 RQ 5.1: Which Language Model?

Firstly, we evaluate the classification performances[8] of the Emb-BiLSTM models (rows 3-7 in Table 5.2) and the BERT-related models (rows 8-12 in Table 5.2). Table 5.5 shows that all of the Emb-BiLSTM and BERT-related models outperform the baseline model. They also outperform LR(BM25) in terms of both BAC and accuracy on the *agree* & *disagree* classes. However, the BERT-based models marginally outperform the Emb-BiLSTM models. We postulate that this is because learning an embedding on a small dataset results in biased textual embeddings for representing terms.

[8]We test our models on a portion of the training data, since the authors of the WSDM 2019 Cup Fake News Challenge dataset never publisehd the ground truth of the test set. Hence, we do not compare our results to the winning group in the WSDM 2019 Cup Fake News Challenge.

Table 5.6: Case study with two examples from the WSDM 2019 Cup Fake News Challenge dataset, where both the LR(BM25) model and the BERT model give the wrong prediction, but our ensemble model gives the correct prediction. *Zh* denotes the text is in Chinese.

Title1 (Zh)	Title2 (Zh)	Title1 (Eng)	Title2 (Eng)	LR(BM25)	BERT	Ensemble	True label
10个孩子空腹吃荔枝死亡？医生的呼吁为所有人敲响警钟！	热传空腹吃荔枝会致人死亡，哈尔滨专家辟谣	10 children died after eating lychees on an empty stomach? Doctors call for everyone to be alarmed!	eating lychee on an empty stomach can lead to death? Harbin doctor debunk the rumor.	*Agree*	*Unrelated*	*Disagree*	*Disagree*
2018年后，农村将“统一住房”，两项补贴10万元	2018年农村要统一修建新房子！直接拎包入住！农民有福了！	Afte 2018, government will provide social housing to countryside families, as well as two subsidies worth ¥100k.	In 2018, the government will build social housing for villegers to move in directly! Good news for farmers!	*Unrelated*	*Unrelated*	*Agree*	*Agree*

Meanwhile, we observe that ECC (accuracy of 0.789) outperforms ECM (accuracy of 0.752) while ESC (accuracy of 0.703) outperforms ESM (accuracy of 0.696). Moreover, ECM and ECC outperform ESM and ESC, respectively. Therefore, for the Emb-BiLSTM-related models, we conclude that the cosine similarity performs better than subtraction, and that using a CNN classifier performs better than MLP. On the contrary, the performances of the two similarity methods used with the BERT-related models are the opposite of that using Emb-BiLSTM (i.e., BSM/BSC outperform BCM/BCC). We do not observe the same performances with the MLP and CNN methods, as BSM outperforms BSC, but BCC marginally outperforms BCM.

Of all the models presented in Table 5.5, the BERT model achieves the best accuracy and F1 score. However, the BSM model achieves the best BAC, as well as the best accuracy on the *agree* and *disagree* classes. Therefore, in response to **RQ 5.1** and **Limitation L1**, we conclude that the BSM model most accurately predicts the *agree* and *disagree* classes in this Chinese news title relationship classification task.

5.4.2 RQ 5.2: Does BM25 Help?

Now, we turn our attention to **RQ 5.2**. Table 5.5 shows that the BSM ensemble model with BM25 achieves the best BAC score (0.847), and the best *agree* and *disagree* class accuracies (0.892 and 0.826, respectively). Indeed, the BACs of all models increase when BM25 is ensembled, but the accuracy scores do not increase consistently. Specifically, the performances of the ESC, ECM, BERT, and BSM models increase significantly when ensembled with BM25 score, compared to the use of the respective models only.

The observation of increasing BAC is particularly interesting, as the BM25 model alone does not achieve a high BAC, but assists other models to perform better for the *agree* and *disagree* classes. Indeed, Table 5.6 presents examples where both the BERT model and the LR(BM25) model predict incorrectly, while the ensemble model predicts correctly.

Table 5.7: Classification scores for the MM-COVID dataset. **Bold** denotes the best result in the table. †† denotes that an ensemble model significantly outperforms both the corresponding RNN/BERT model as well as LR(BM25) (McNemar's test, $p < 0.01$).

Model	Acc	BAC	P	R	F1	Fake Acc	Real Acc
BL	0.892	0.885	0.88	0.88	0.88	0.864	0.905
LR(BM25)	0.906	0.901	0.89	0.90	0.90	0.886	0.916
ESM	0.898	0.891	0.88	0.89	0.89	0.870	0.912
+ BM25††	0.905	0.901	0.89	0.90	0.90	0.886	0.915
ESC	0.901	0.894	0.89	0.89	0.89	0.873	0.915
+ BM25††	0.908	0.904	0.89	0.90	0.90	0.889	0.918
ECM	0.903	0.898	0.89	0.90	0.90	0.880	0.915
+ BM25††	0.906	0.904	0.89	0.90	0.89	0.898	0.910
ECC	0.915	0.911	0.90	0.91	0.91	0.898	0.924
+ BM25	0.914	0.912	0.90	0.91	0.91	0.907	0.918
BERT	**0.945**	**0.942**	**0.94**	**0.94**	**0.94**	0.934	**0.950**
+ BM25††	0.940	**0.942**	0.93	**0.94**	0.93	**0.949**	0.935
BSM	0.934	0.931	0.92	0.93	0.93	0.919	0.943
+ BM25	0.930	0.930	0.92	0.93	0.92	0.931	0.930
BSC	0.931	0.930	0.92	0.93	0.92	0.928	0.933
+ BM25††	0.933	0.935	0.92	0.93	0.93	0.940	0.930
BCM	0.920	0.917	0.91	0.92	0.91	0.907	0.927
+ BM25	0.919	0.920	0.91	0.92	0.91	0.922	0.918
BCC	0.920	0.918	0.91	0.92	0.91	0.910	0.926
+ BM25††	0.922	0.924	0.91	0.92	0.91	0.928	0.919

Therefore, regarding **RQ 5.2** and **Limitation L2**, we conclude that including the BM25 scores does improve the performances of using the classifiers with the neural network language models, especially improving the performances on the *agree* and *disagree* classes.

5.4.3 RQ 5.3: Identifying Recurring Fake News

Table 5.7 presents the classification results of each tested model on the MM-COVID dataset. We observe that the BERT language model achieves the highest accuracy (0.945), BAC (0.942), and F1 score (0.94) in identifying recurring fake news, among all tested models. However, BERT + BM25 achieves the highest accuracy on the fake news class, significantly better than the BERT model alone. Similarly, although combining with the BM25 model decreases the overall accuracy for ECC,BERT, BSM, and BCM, the accuracy on the fake news class increases consistently over all the language models, albeit not always significantly. Moreover, we observe inconsistent results in terms of BAC. That is, the BAC scores increased for all language models except the BERT and the BSM models, when ensembled with BM25, where BSM+BM25 has insignificantly lower BAC than the BSM model, while the BERT model achieves the same BAC as BERT+BM25.

This observation echoes the findings from RQ 5.2, which also shows that the BM25 scores

Table 5.8: Case study with two examples from the MM-COVID dataset, where both the LR(BM25) model and the BERT model give the wrong prediction, but our ensemble model gives the correct prediction.

Check-worthy claim	Debunked fake news	LR(BM25)	BERT	Ensemble	True Label
deficiency oxygen fatigue prolonged cause masks	Wearing masks for the coronavirus "decreases oxygen intake increases toxin inhalation shuts down immune system increases virus risk scientifically inaccurate effectiveness not studied.	Real	Real	Fake	Fake
Dolores Cahill nutrition vitamins	Dolores Cahill claims in an interview on TheHighWire that there is already a preventive strategy and treatment for COVID-19 through "nutrition vitamins and hydroxychloroquine	Fake	Real	Fake	Fake

assist the language models in more accurately identifying the *agree* and the *disagree* classes, than using the respective language models alone. We postulate that when a check-worthy tweet/sentence is being compared with existing fake news, the BM25 scores can help distinguish the *unrelated* cases more easily, thus reducing the potential errors language models make in misclassifying *unrelated* cases as the *agree* cases. Indeed, Table 5.8 presents two examples, where the first example shows a case where both the BERT model and the LR(BM25) model predict incorrectly, while the ensemble model predicts correctly. We can observe from this case that the two sentences do not have many overlapping words. Thus, LR(BM25) is not very effective, whereas the semantic similarities between the two sentences are not strong either, due to the check-worthy claim's keywords style of presentation. However, the ensemble model can combine the BM25 similarity and semantic analysis to predict the check-worthy claims as fake. On the other hand, the second case presented shows that the BM25 score can overpower the BERT model, and the ensemble model can correctly predict the check-worthy claims as fake.

Therefore, regarding **RQ 5.3**, we conclude that including the BM25 scores does improve the classification performance of the language model based-classifiers, in identifying recurring fake news.

5.5 Conclusions

In this chapter, we addressed a core task needed for fake news detection, which aims to leverage an existing fake news collection in identifying recurring fake news. In particular, we investigated various neural network-based language representations for detecting the relationships between check-worthy claims/tweets and debunked fake news, such as BiLSTM approaches, BERT-based approaches [105], and the jointly trained BERT model and BM25 approaches [105].

Our thorough experiments (presented in Section 5.4.1) showed that using BERT for text representation, using the subtraction similarity method and MLP as the classifier predicted the *agree* and *disagree* classes most accurately among tested language model approaches. Moreover, Section 5.4.2 showed that, when the neural network language models are combined in an ensemble manner with the BM25 similarity, it resulted in improvements (albeit not always significant) to the effectiveness of all language model approaches. Finally, in Section 5.4.3, we showed that such classification models are useful in identifying recurring fake news, using the MM-COVID dataset, where we aimed to identify fake news from a set of claims, by comparing each claim with the set of existing fake news. Specifically, we showed that the BERT model performs the best in terms of accuracy and F1 score, in the recurring fake news detection task. Furthermore, the BM25 model again can improve all tested language models in terms of accuracy on the **fake news class**.

In answering RQs 5.1 & 5.3 and **Limitation L1** (Section 2.6.4), Table 5.5 demonstrated that the BSM approach (i.e., BERT model, subtraction similarity, and MLP classifier) outperforms other tested approches, in classifying the relations between news titles, while Table 5.7 showed that the BERT model outperforms other language model approaches in identifying recurring fake news. Thus, we conclude that the BERT model is the most suitable language model representing news titles, tweets and claims, in identifying recurring fake news.

In answering RQ 5.2 & 5.3 and **Limitation L2** (Section 2.6.4), Table 5.5 & 5.7 showed that the BM25 model can aid language models, in accurately classifying the *agree* and *disagree* relations between two news titles, and further helps language model to more accurately predict the fake news class.

In answering RQ 5.3 and **Gap 3** (Section 2.4.4), Table 5.7 showed that an ensemble model of BM25 and the BERT language model can effectively identify recurring fake news, by comparing tweets and claims needed to be fact-checked with previously debunked fake news, compared to using either the BM25 scores or the BERT language model alone.

These findings suggest that a BM25 matching score can aid neural language model approaches, and ensemble methods can perform better than each component used alone, arguably because BM25 can better identify similarities that are difficult for the language mod-

els to learn. Thus, in addressing the hypothesis presented in the thesis statement in Section 1.3, we conclude that our proposed ensemble model of the BM25 scores and language representations can accurately classify if a targeted check-worthy claim is highly similar to any existing fake news, and thus is a resurfaced fake claims.

We note that when identifying recurring fake news using existing fake news dataset, both the quality and the quantity of existing fake news are important in achieving better performance in identifying recurring fake news. Thus in practical settings, it is important to obtain an extensive fake news dataset.

Finally, we summarise the contributions of this chapter as follows:

1. Comparing simple-embedding representations, BiLSTM and BERT, we draw best practices in using BERT language model representations in classifying the relationship between Chinese news titles, and between claims/tweets and debunked fake news.

2. We showed the traditional BM25 retrieval scores can improve the performance of deep neural network models, such as the BiLSTM model and the BERT model.

3. In answering RQ 5.3, Table 5.7 showed that the NLI relations between tweets/claims and previously debunked fake news can effectively identify recurring fake news from a set of check-worthy tweets and claims.

In the next chapter, we discuss our proposed model in identifying fake news by leveraging social media networks.

Chapter 6

Social Network Structure Assisted Fake News Detection

6.1 Introduction

In Section 2.1, we provided an overview of the role social media platforms play in the spreading of fake news online. We surveyed research that aim to study the spreading pattern of news on social media platforms [28, 102], and recognised that repeated exposure of news could make users believe in what their relatives and friends have shared [193, 204]. Yoo [199] identified the *echo chamber* effect as one of the main reasons that fake news is prominent on social media platforms, as like-minded people tend to gather in small groups, where others in the same group confirm their existing beliefs, and amplify each other's opinions.

To combat the rapid spread of fake news on social media platforms, Section 2.4.3 surveyed several methods to leverage social media platforms' information in fake news detection, where both static features and social network embeddings are used in fake news detection on Twitter. For example, numeric features such as numbers of followers, verified or not, and user descriptions [97] are used as hand-crafted features in fake news detection. The relations between users engaging in the same news can also be useful to determine the truthfulness of the news, such as if two users follow each other, in the same region, engaged with the same tweet/URL [152]. Moreover, the propagation of tweets can be an important attribute in identifying non-factual information on Twitter, such as replies, retweets, likes, viewpoints conflicts of a tweet in need of fact-checking [75]. Similarly, some researchers [128, 147, 165] aimed to study the users' connection on social media and leverage the users' connection to detect fake news. For example, the network of YouTube channels and individuals propagating fake news is usually integrated with heterogeneous discussion networks that involve factual content more than misinformation [147]. On the Twitter platform, Nguyen et al. [128] proposed the Factual News Graph model (FANG) to use embedded network information

to identify untrustworthy news stories. Furthermore, Sosnkowski et al. [165] showed that changes in the users' network structure on Twitter could help detect the change in political opinions among users.

Section 1.3 presented our thesis statement, where we hypothesised that in the fact-checking phase (Phase 2), **user network embeddings** trained with **unlabelled** user network data, can identify the echo chamber effects among users, and is effective in identifying fake claims on Twitter. This chapter aims to test this hypothesis, by leveraging the users' network structure on Twitter to build a model that effectively identifies fake news on Twitter. That is, we aim to tackle Task 3 (defined in Section 3.3.3), to determine whether the content of a tweet is truthful or not, using the Twitter user's user information. In this chapter, we propose a novel fake news detection model that uses network embeddings to accurately identify fake news on Twitter. In particular, we propose the User Network Embedding Structure (UNES) model to represent each Twitter user in a lower-dimensional space, based on their connections with other users within the platform, and projecting the social network structure as a linked graph. We argue that this graph structure can aid in detecting clusters of users engaging in fake news and assist the fake news detection task on Twitter.

This chapter aims to address **Gap 4** identified in Section 2.4.4, which states that the social media users' connections with each other are largely overlooked in fake news detection systems. Thus, this chapter addresses **Limitations N1 and N2** introduced in Section 2.7.2.2, which elicit on **Gap 4**.

Limitation N1 identifies the need to learn user embeddings from the readily available network information in order to conduct large scale fake news detection on Twitter, compared to using complex network structure that requires labourious preprocessing. To address **Limitation N1**, we propose to build an extensive user network, using only users' followers and friends, which can be easily acquired using the Twitter API. Our experiments show that our user embeddings learnt using unsupervised models can outperform the SOTA model [128] that uses a sophisticated hand labelled network.

Limitation N2 aims to find the most effective type of users connections to use in the construction of a user network in detecting fake news on Twitter. To address this limitation, we propose to compare the user networks constructed with the users' friendship relations against the user network constructed with the users' follower relations. Our experiments show that the users' friendship network is more effective at separating users engaged with fake news, and helps identify fake news more effectively, than the users' follower network.

We test our proposed UNES on the SD datasets [128]. Our experiments show that the unsupervised user network embeddings can indeed separate users into different groups, based on their engagement with fake news. Furthermore, our experiments show that our proposed UNES model using user embeddings obtained from the users' friendship network signifi-

cantly outperforms the SOTA model, which in-turn significantly outperforms the language models based- fake news detection methods. Thus, in answering the hypothesis presented in the thesis statement in Section 1.3, we show that our proposed user embedding assisted fake news detection model UNES can effectively classify whether a targeted check-worthy claim is fake or not based on the engaged user embeddings.

This chapter is structured as follows: Section 6.2 formally states the task to be tackled and describes our proposed UNES model for fake news detection; The experimental setup and obtained results for fake news detection using the UNES model are provided in Sections 6.3 and 6.4, respectively; Concluding remarks follow in Section 6.5.

6.2 Using Social Network Embedding for Fake News Detection (Phase 2 Task 3)

This section builds upon the objective of Task 3 presented in Section 3.3.3, describes the task we aim to tackle, and introduces our proposed UNES model in detail.

6.2.1 Twitter Users-Based News Article Classification

In order to develop a model that allows accurate classification of a claim/tweet/news x as fake or not, we propose to calculate the truthfulness of a given claim/tweet based on its engaging tweets, and the corresponding users that posted the engaging tweets.

Table 6.1 shows the set of notations (a subset of notions defined in Section 3.2, and a set of newly defined notations) we use in this chapter.

Table 6.1: Notations used in Chapter 6.

Notation	Definition
$X_{checkworthy}$	The set of check-worthy claims and tweets
x	A sentence or tweet in the set of sentences and tweets $X_{checkworthy}$
T_x	The set of tweets related to x
T	The set of tweets of all T_x for $\forall x \in X_{checkworthy}$
t	A tweet
U_x	The Twitter users that engaged with x, who posted the tweets T_x
U	The set of users who posted the set of tweets T
u	A Twitter user who posted t
G	The graph that consists of users U and their friends or followers on Twitter
$\overrightarrow{t}$	The modelled vector representation of tweet t
$\overrightarrow{u}$	The modelled vector representation of user u
a_x	The text content of x

Recall the definition of Task 3 presented in Section 3.3.3:

$$\widehat{Y_x} = cls(x, T_x, U_x) \tag{6.1}$$

The main objective of this task is to classify whether a claim/tweet/news article x is fake news or not, given the engaged tweets T_x and engaged users U_x of x. We expand the definition of the classification task as follows:

$$\widetilde{Y_x} = cls(x) = cls(a_x, T_x, U_x) \tag{6.2}$$

In particular, for each news article/claim/tweet x requires fact-checking, we aim to predict whether x is fake or not, $\widetilde{Y_x}$, based on the text of the news article/claim/tweet a_x, the tweets T_x that engaged with x (e.g. tweeted the same news/claims; retweeted/commented on the news/tweet), as well as the users U_x that tweeted T_x, where $u \in U_x$ posted the tweet $t \in T_x$. Hence, the objective of this study is to identify the best $cls()$ for fake news detection using such information – next, we describe our proposed UNES model, which identifies fake news on Twitter based on the network structure, G, of Twitter users.

6.2.2 Proposed Model - UNES

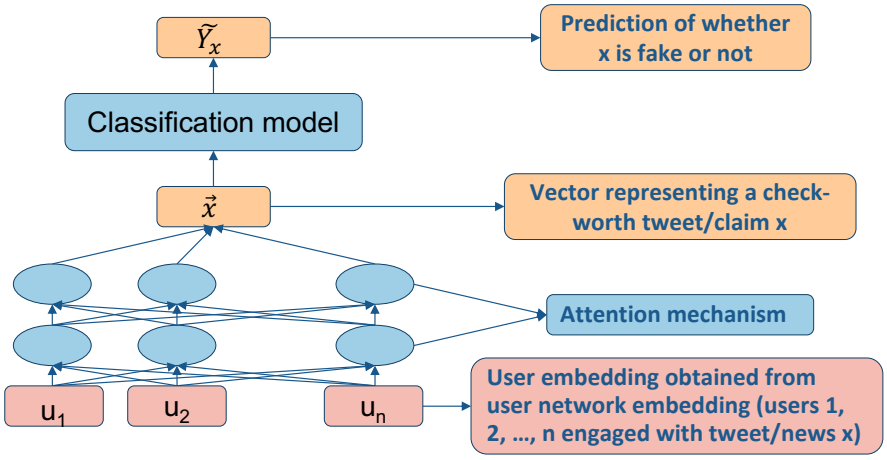
We now describe our proposed User Network Embedding Structure (UNES) model for fake news detection on Twitter, which represents engaging users with their user embeddings obtained from network embedding methods. To demonstrate our proposed UNES, Figure 6.1(a) presents the structure of the UNES model, while Figure 6.1(b) shows a BERT-based model as an example of the state-of-the-art language model for fake news detection.

To obtain a prediction for a given claim/tweet/news article x, we make use of the social network connections of each engaged users $u \in U_c$. We consider each user as a node in a directed graph, and their relationships (i.e. following, friendship) as vertices connecting with other users. To this end, we introduce $\text{follows}(u_i, u_j)$ as a binary function that returns 1 if user u_i follows u_j, and 0 otherwise. Moreover, Twitter users can be friends – i.e. follow each other – for which we use two edges that have different directions but connecting the same two nodes: $\text{follows}(u_1, u_2) \wedge \text{follows}(u_2, u_1)$.

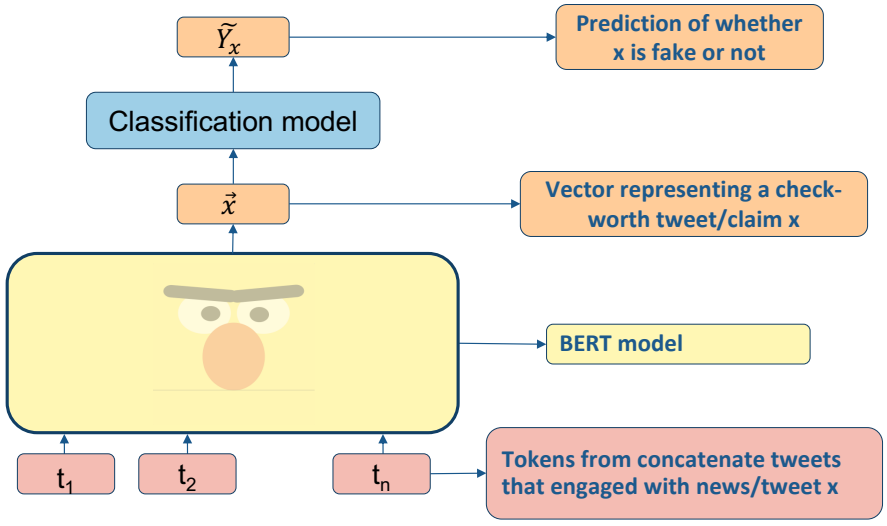
For all users U, the social network graph can be expressed as an adjacency matrix G. Specifically, let G^{fo} denote the follower graph, and G^{fr} the friend graph. Entries in these adjacency matrices for any pair of users $u_i, u_j \in U$ are defined as $\text{follows}()$:

$$G^{\text{fo}}_{i,j} = \text{follows}(u_i, u_j) \tag{6.3}$$

$$G^{\text{fr}}_{i,j} = \text{follows}(u_i, u_j) \wedge \text{follows}(u_j, u_i) \tag{6.4}$$



(a) Our proposed UNES model for fake news detection.



(b) Language model (BERT) baseline for fake news detection.

Figure 6.1: Comparison between our proposed UNES model and a text-based (BERT) baseline classifier.

In the following, we use G to represent either G^{fo} or G^{fr}.

To make use of the large social network graphs within a classifier, we convert the sparse network structure into dense graph embeddings. In particular, the connections of u are defined as all users that user u is connected to (i.e., as either a follower or a friend). Thus, we represent each user u by their connections to other users through the application of a graph embedding function[1] $f_u()$, to obtain embeddings for each user u, as follows:

$$\overrightarrow{u} = f_u(G_u) \tag{6.5}$$

[1]In the remainder of this chapter, we will use graph embedding and network embedding interchangeably, unless it is required otherwise by the context of the section, as we apply graph embedding to the user network structure.

Table 6.2: Statistics of the SD dataset. Note that the avg., max., and min. numbers are per news article.

	Overall	Fake	Factual
# of news	1054	448	606
# w/o tweets	11	4	7
avg. # tweets	46.30	52.31	41.89
max. # tweets	1054	750	1054
avg. # users	34.10	47.48	35.95
max. # users	1028	645	1027

Table 6.3: Statistics of the edges users have in our friendships and follower networks, for the SD dataset[2]. $\# > 5000$ denotes the users with more than 5000 friends/followers shown in their profile.

	# edges of G^{fr}	# edges of G^{fo}
Avg	1388.37	1565.16
Max	14012	18255
Std.	1614.43	1891.05
25%l	181	147
50%l	674	621
75%l	2017	2498
$\# > 5000$	3808	6263

Note that we employ unsupervised graph embedding models. Specifically, users are not labelled with their engagement with fake news or factual news. This is a realistic setup, since a large number of users may not have engaged with any news articles on Twitter, fake or factual.

In our proposed UNES, we represent each news article, by applying the multi-head attention mechanism [184] on all the engaged users' embeddings (i.e., users who tweeted about x) to capture the rich information among a group of users. Thus, Equation (6.2) can be instantiated as follows:

$$\widetilde{Y_x} = cls(x) = cls(\underset{u \in U_x}{\text{Attention}}(f_n(G_u))) \tag{6.6}$$

where a_x and T_x are optional in the classification function $cls()$, and Attention is the multi-head attention function.

In particular, we use three types of graph embedding methods for the fake news classification task, representing the basic graph embedding models (i.e., DeepWalk), graph cluster-based graph embedding models (i.e., Cluster-GCN), and the existing state-of-the-art graph embedding models (i.e., GraphSAGE), namely:

1. **DeepWalk** [136]. This was the first deep learning-based method to address a graph

[2]10 users missing, as 2 accounts are set as private, and 8 accounts were deleted.

embedding task, in a manner inspired by Word2Vec [117]. DeepWalk takes truncated random walks on a graph to learn embedded representations of nodes.

2. **Cluster-GCN** [82]. Given a graph, Cluster-GCN uses a graph convolution operation (GCN) [82] to obtain node embeddings, by aggregating the neighbouring nodes' embeddings of each node, applying CNN layers for each aggregation. Moreover, Cluster-GCN identifies a subgraph using a graph clustering algorithm, and restricts the neighbourhood search within this subgraph, thus presenting a more efficient and effective graph embedding model than a plain GCN.

3. **GraphSAGE** [68]. This method performs parameterised random walks and uses recurrent aggregators. It can be used for both unsupervised and supervised representation learning, and it can generate embeddings for unseen nodes and edges at inference time.

Thus, using the graph embeddings from these three methods, we can represent a news article in a hyper-dimensional space based on the social network structure of its engaged users (aggregated using the multi-head attention mechanism as per Equation (6.6)), without needing textual information, or a more sophisticated analysis of each users' account (e.g., account type, stance on the topic). Furthermore, in order to compare the effectiveness of our UNES model with widely-used textual features for detecting fake news on Twitter, in our experiments, we use classifier models learned using social network graph embedding features (as in Figure 6.1(a)), as well as those using textual features (as in Figure 6.1(b)) – such as the state-of-the-art language model BERT. [40] Next, we detail our research questions and the setup for our experiments.

6.3 Experimental Setup

We address three research questions as follows:

- **RQ 6.1:** Can our proposed UNES model allow to identify clusters of users who engage with fake news on Twitter?

- **RQ 6.2**: How effective is UNES in identifying fake news on Twitter and which graph embedding method is the most effective? This RQ aims to address **Limitation N1** presented in Section 2.7.2.2, which concerns the effectiveness of using user embeddings in the detection fake news, and the effectiveness of each graph embedding model.

- **RQ 6.4:** Which type of social network structure (i.e., followers or friendship networks) provides the most effective information in allowing to accurately identify fake news on

Twitter? This RQ aims to address **Limitation N2**, which concerns the identification of the most effective type of users connections in detecting fake news on Twitter.

In the following, we describe the used dataset, the approaches we use to represent both tweets and news articles, the user network embedding models, the baselines, and the evaluation metrics we use in reporting our results.

6.3.1 Dataset

In our conducted experiments, we use the stance detection (SD) dataset provided by Nguyen et al. [128][3]. The SD dataset consists of news article links and human judgements labels denoting if they are fake or not, as well as engaged tweets, the stance of such tweets, the publisher of the news article, and article citations by other news outlets on Twitter. We download all the available tweets in the dataset, along with the tweet authors' friends and followers. We limit this to a maximum of 5000 for friends and 5000 for followers, which is the maximum number we can download as per the limit of Twitter API calls. We also remove those users who are connected to only five other user in the graph (provided they have not engaged with any news article) in order to reduce the size of the graph and allow for tractable experiments. In Table 6.2 we provide the statistics of the dataset in terms of news articles, tweets and users; Table 6.3 details statistics of the friendship and follower networks of the users. One can argue that information such as likes, replies, and retweet relations can also be viewed as possible types of relationships on Twitter. However, due to the difficulties in obtaining data concerning likes, replies, and retweet relationships from the Twitter API, we do not use these types of relationships in our work.

6.3.2 Semantic Representation

In order to investigate the effectiveness of social network structure in detecting fake news, we also deploy textual-based classifiers as baselines. In particular, we employ two language processing methods, namely, a TF.IDF representation and a BERT representation for both tweets and news articles. The experimental setup for these two language processing methods are as follows:

- **TF.IDF:** We use NLTK's TweetTokenizer[4] to tokenise tweets. Scikit-Learn's TfidfVectorizer[5] is used to extract the TF.IDF representation for both news content a_x and

[3] https://github.com/nguyenvanhoang7398/FANG
[4] https://www.nltk.org/api/nltk.tokenize.html
[5] https://scikit-learn.org/

tweets T_x. We limit the maximum number of tokens per text entry to 10k, to include all the tweets and news article tokens.

- **BERT:** We fine-tune the BERT-base English model (uncased, 12-layer, 768-hidden, 12-heads, 110M parameters) using the validation set. We maintain the suggested learning rate [40], with a drop out rate of 0.05. The maximum token length for the concatenation of news article and engaged tweets is 512 (i.e., as the maximum number of tokens that BERT can encode is 512 [40]).

6.3.3 User Network Embedding Methods

As mentioned in Section 6.2.2, we instantiate the UNES model with three graph embedding methods, namely DeepWalk, Cluster-GCN, and GraphSAGE. Each of these graph embedding methods allows two sets of features per node: node network structures and additional node features. In order to address RQ 6.3, we deploy our models without textual node features. The only information we pass to the graph embedding methods are the network connection features. Moreover, the two types of relationships we use to initiate the graph embedding methods are *friendship* and *followers*. The detailed setup of each graph embedding method is as follows:

- **DeepWalk.** We train DeepWalk with 50 hidden units, a window unit of 10. Each node has a maximum of 10 walks, with a maximum of 80 steps per walk, and results in a 64 dimension vector to represent each user, as per the original DeepWalk paper [136].
- **Cluster-GCN.** We train Cluster-GCN with 1000 epochs each, with 16 hidden units, and results in a 100 dimension vector to represent each user, as per Nguyen et al. [128].
- **GraphSAGE.** We train GraphSAGE with 30 epochs, 16 hidden units, and 2 layers. It results in a 100 dimension vector for each user, as per Nguyen et al. [128].

Note that we represent the users who do not have any followers or friends using an embedding vector of [-1, ..., -1].

6.3.4 Classifiers

We use SVM as our baseline classifier model. This model uses the TD.IDF model to represent the article content a_x and tweets T_x concatenation. We use the Scikit-Learn's implementation of SVM, with the standard parameters (i.e., RBF kernel, a C penalty of 10, and a γ of 0.1). For all the deep learning models (both network based and language based), we use a

fully connected dense layer to classify a news article as fake or factual, which is trained end-to-end with the user embedding vectors. For instance, for the BERT model, we fine-tune the pre-trained BERT model with a fully connected dense layer; for our proposed UNES model, we train the fully connected dense layer as $f_n()$, as per Equation (6.2).

6.3.5 Baselines

We report two groups of baselines, 5 baselines in total – in order to test the effectiveness of our UNES model.

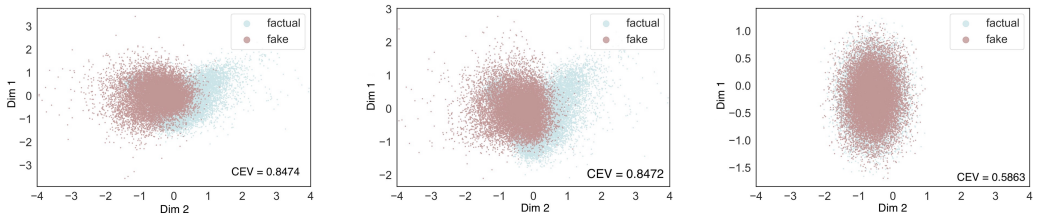
The first group of baselines consists of models that only use textual features. The second group of baselines are the current state-of-the-art models proposed by Nguyen et al. [128], namely the GCN model and GraphSage model using the FANG network. These five baselines are as follows:

1. Textual features only:
    1.1. SVM classifier with a TF.IDF representation of tweet and news article, denoted as **SVM TF.IDF(a_x and t_x)**;
    1.2. Fine-tuned BERT model, using the news content a_x, denoted as **BERT(a_x)**;
    1.3. Fine-tuned BERT model, using the content of the engaged tweets T_x, denoted as **BERT(T_x)**;
2. FANG Models proposed by Nguyen et al. [128]:
    2.1. GCN with FANG network information, as well as TF.IDF representation for the tweets and news, denoted as **GCN (G^{FANG})**;
    2.2. GraphSage model using FANG network information, as well as TF.IDF representation for the tweets and news, denoted as **GraphSage(G^{FANG}**.

6.3.6 Evaluation Metrics

We evaluate our methods using the SD dataset, with existing training and testing splits, where the training set is 10%, 30%, 50%, 70%, and 90% of all data, provided by Nguyen et al. [128]. Hence, we test all the models' performances using the same testing sets, and the performances are therefore comparable to those numbers reported by Nguyen et al. [128].

As evaluation metrics, we report macro Accuracy, Precision, Recall, and F1, as well as Area under the ROC Curve (AUC) – indeed, we go further than previous work on the SD dataset, which focused solely upon AUC. Moreover, in order to visualise the effectiveness of the



(a) Friendship network. (b) Follower network. (c) Random user network.

Figure 6.2: Unsupervised embedded users shown in PCA mapping. Figures 6.2(a) and 6.2(b) are trained using DeepWalk, while Figure 6.2(c) uses randomly generated user network embeddings.

users' embeddings in identifying groups of people engaged with fake news versus factual news, we apply the PCA dimension reduction technique to the users' embeddings. For the PCA technique, we also report the cumulative explained variation (CEV), which analyses the variation for individual components, and sums up the variation of the k ($k = 50$ in our experiments) most varied principal components. This metric shows the variation between two groups of users (i.e., users that have engaged with fake news vs. users that have never engaged with fake news) that are embedded using the friendship/follower networks.

6.4 Results and Analysis

In this section, we present the results from our experiments to answer our three research questions presented in Section 6.3. Furthermore, we conduct a case study and discuss the implications of the obtained results.

6.4.1 RQ 6.1: Clustering Effect of Users' Network Embeddings

To gauge the effectiveness of unsupervised user network embeddings in identifying clusters where users engage with fake news, we visualise the distributions of users in our embedded user networks, using the PCA dimensionality reduction technique[6], and measure the cumulative explained variation (CEV) between users who engage with fake news and users who engage with factual news. Specifically, we visualise the users' embeddings trained with the DeepWalk graph embedding method, since DeepWalk provides a lower bound in user embeddings performance, due to its simplicity compared to the Cluster-GCN and the GraphSAGE methods.

[6]Plots using the tSNE dimensionality reduction technique produced similar observations, and hence are omitted for brevity reasons.

Figures 6.2(a) and 6.2(b) illustrate the distribution of user embeddings (learned from the friendship network and the follower network with the DeepWalk graph embedding method) when reduced to 2 dimensions using a PCA. The red dots denote users that have engaged with at least one fake news article, while the teal dots denote users that **only** engaged with factual news articles. Note that as mentioned earlier, during the training session, we did not label users as engaged with factual news or engaged with fake news, hence the embeddings are learned in an unsupervised fashion. In Figure 6.2(c), we also show randomly generated user embeddings plotted using PCA, where, unlike in Figures 6.2(a) and 6.2(b), the distributions of users engaging with fake news and with factual news are more uniform.

Specifically, from Figure 6.2(b), it can be observed that for the user embeddings learned from the friendship network, users who are engaged with fake news are more tightly clustered together. That is, the PCA mapping shows that users who engaged with fake news are tightly gathered around the top left corner, which suggests that the users who engage in fake news are more tightly grouped into smaller echo chambers than the users who do not engage in fake news. This echoes the findings of Yoo [199], who coined the term echo chamber. Moreover, the CEV analysis shows that the top 50 principal components achieved a cumulative explained variation of 0.8474, which indicates that the two groups of users have a variance of 0.8474 from the embeddings learnt from the friendship network. In Figure 6.2(b), we observe similar trends from the user embeddings learned from the follower network, with the CEV for the 50 principal components being 0.8472, which indicates that there are valid differences between the two user groups, in comparison to the randomly generated user embeddings, where the CEV is 0.5863. Our results echo the findings reported by Törnberg [178], namely that users tend to form a more tightly grouped community when they are more heavily influenced by fake news, and show that echo chamber effects indeed exist in social media.

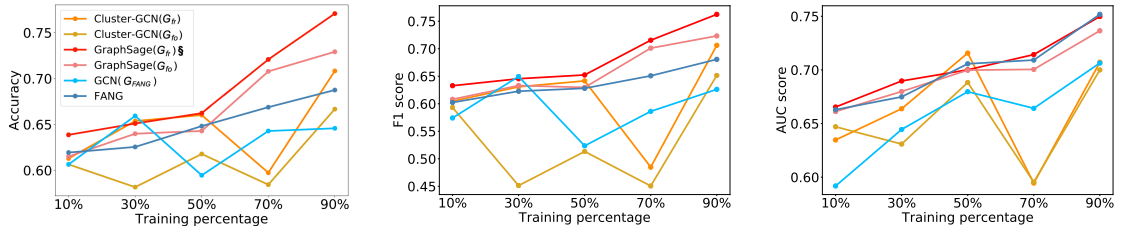
Thus, in response to RQ 6.1, we conclude that the unsupervised Twitter user network embeddings can indeed cluster users into different groups (i.e., users who have engaged with fake news, versus users who only engaged with factual news), with a cumulative explained variation of around 85%, using either the follower network or the friendship network. Recall thesis statement in Section 1.3, where we hypothesised that the unsupervised user network embedding can help identify the echo chamber effects among users. Based upon the experiments shown in this chapter, we conclude that unsupervised user network embedding can indeed show users grouping among Twitter users.

6.4.2 RQ 6.2: UNES Model for Fake News Classification

To address RQ 6.2, we compare our unsupervised user network embeddings with language models and user embeddings trained from sophisticatedly labelled social networks FANG.

Table 6.4: Performance comparison among the models using 90% training data. Numbers in the significance column indicates that the model is significantly better than the numbered model (McNemar's Test, $p<0.01$).

#	Model	Accuracy	P	R	F1	AUC	Significance
Textual Baselines – Textual features only							
1	Random	0.4737	0.3205	0.5000	0.3906	0.5000	-
2	SVM TF.IDF(a_n and T_n)	0.6068	0.6010	0.6095	0.5962	0.6095	1, 4
3	BERT(a_n)	0.5897	0.5584	0.5595	0.5588	0.5595	1, 4
4	BERT(T_n)	0.5299	0.5410	0.5443	0.5249	0.5443	1
FANG models– Complex network features and texutal features (publisher, citation, follower network, stance and TF.IDF features)							
5	GCN(G^{FANG})	0.6458	0.6328	0.6250	0.6262	0.7125	1-4, 7
6	GraphSage(G^{FANG})	0.6875	0.6799	0.6821	0.6807	**0.7518**	1-5, 7,9
UNES variants – Unsupervised network features only							
7	DeepWalk(G^{fr})	0.6410	0.5717	0.5052	0.4122	0.5052	1-4
8	Cluster-GCN(G^{fr})	0.7083	0.7083	0.7142	0.7062	0.7071	1-7,9
9	Cluster-GCN(G^{fo})	0.6667	0.6556	0.6500	0.6515	0.7000	1-5,7
10	GraphSAGE(G^{fr})	**0.7708**	**0.7650**	**0.7607**	**0.7625**	0.7498	1-9,11
11	GraphSAGE(G^{fo})	0.7292	0.7222	0.7250	0.7233	0.7365	1-9



(a) Accuracy. (b) F1 score. (c) ROC AUC.

Figure 6.3: Performances of GraphSAGE(G^{fr}), GraphSAGE(G^{fo}), Cluster-GCN(G^{fr}), Cluster-GCN(G^{fo}), GCN(G^{FANG}), and the FANG model, on accuracy, F1, and AUC. § in the legend denotes that our corresponding model variant significantly outperforms FANG, on all training percentages.

Specifically, we evaluate whether our proposed UNES model is more effective in detecting fake news on the SD dataset w.r.t. the baseline models. We also identify the most effective graph embedding method for fake news detection on Twitter, by comparing our UNES using various instantiations.

Table 6.4 shows how the models perform when trained with 90% of all data, validated with 5% of the data, and tested on 5% of the data. Indeed, we use the pre-partitioned training-testing sets provided by Nguyen et al. [128], thus the results shown in Table 6.4 are comparable to the GCN(G^{FANG}) and the GraphSage(G^{FANG}) models (row 5 and 6 in Table 6.4). To analyse the effect of the size of the training data on the results, we also evaluate the various models across different training data size percentages, and present the results in Figure 6.3, in terms of Accuracy, F1 and ROC AUC.

On analysing Table 6.4, we firstly observe that the text-based baselines (i.e., the TF.IDF and BERT models, lines 2-4 in Table 6.4) can identify fake news significantly better than the random baseline. However, a simple SVM(TF.IDF) model outperforms both BERT models, indicating the difficulties for contextual features to analyse news and tweets on Twitter, with the SD dataset. Moreover, all models that use network features (lines 5-11) significantly outperform the textual features only models (lines 1-4). This observation suggests that network features are more successful at identifying fake news on Twitter than textual features alone.

Secondly, we observe that with only the network information, our UNES model, with the simpler DeepWalk graph embedding (line 7 in Table 6.4) does not outperform either of the network baselines (lines 5 & 6). On the other hand, contrary to DeepWalk, the use of the UNES model along the advanced graph embedding models (i.e., Cluster-GCN and GraphSAGE), can achieve better accuracy and F1 performances, compared to the GCN(G^{FANG}) and GraphSage(G^{FANG}), respectively. Indeed, Cluster-GCN with friendship/follower networks (lines 8 & 9) can significantly outperform GCN(G^{FANG}) (line 5), while GraphSAGE with the friendship/follower networks (lines 10 & 11) significantly outperforms GraphSage(G^{FANG}) (line 6). However, GraphSage(G^{FANG}) (line 6) obtained the highest ROC AUC performance among all the tested models. A further inspection on the output of all of the evaluated UNES variants (i.e., lines 7-11 in Table 6.4) shows that while they are accurate, they are generally less certain in their predictions. Specifically, the dense neural network layer outputs posterior probabilities for each class closer to 0.5 rather than 0 or 1 for our binary classification task (i.e., classifying a given news as fake or not), while the GraphSage(G^{FANG}) model tends to produce probability outputs closer to 0 and 1 rather than 0.5. We argue that this is because the UNES model uses a network structure data with no textual information on the content of the news or tweets, unlike GraphSage(G^{FANG}), which uses both the textual data (such as the stance of a tweet and TF.IDF representations of the news articles and the tweets) along with the network data. We leave to future work to investigate of how best to integrate the textual data into the UNES model.

Furthermore, Figure 6.3 shows the performances of the UNES variants and the FANG baselines (GCN(G^{FANG}) and GraphSage(G^{FANG})) when tested with all possible training data size percentages. From the figure, we observe that the UNES variant using the GraphSAGE(G^{fr}) graph embeddings consistently significantly outperforms GraphSage(G^{FANG}) in terms of the accuracy and F1 metrics, regardless of the used training data size. However, similar to the observation that GraphSage(G^{FANG}) achieved the highest ROC scores in Table 6.4, we observe that all the variants of our UNES model (i.e., Cluster-GCN(G^{fr}), Cluster-GCN(G^{fo}), GraphSAGE(G^{fr}), and GraphSAGE(G^{fo})) do not outperform GraphSage(G^{FANG}) consistently, due to the aforementioned issue, namely that model GraphSAGE(G^{fr}) tends to predict posterior probabilities closer to 0.5 rather than 0 or 1.

Moreover, from Figure 6.3 we observe that all the GCN-based models (i.e., Cluster-GCN(G^{fr}),

Cluster-GCN(G^{fo}), and GCN(G^{FANG})) suffer from instability when trained using different percentages of training data. Specifically, the models using the Cluster-GCN(G^{fr}) and the Cluster-GCN(G^{fo}) models both exhibit performance drops when trained with 30% and 70% of the data. One of the baseline models, GCN(G^{FANG}), shows a drop in the Accuracy and F1 performances when trained with 50% of the data, and the ROC AUC drops when trained with 70% of the data. Indeed, the GCN-based models have unstable performances when tested on the different training percentages. This suggests that the GCN graph embedding model might represent nodes (i.e., users in our experiments) inaccurately as embeddings, as GCN and Cluster-GCN both identify subgraphs before computing the embeddings of each node, while the subgraphs are not updated throughout the training session, contrary to the GraphSAGE model, which aims to perform graph embeddings without any subgraph partitioning.

Overall, in response to RQ 6.2 and **Limitation N1**, we conclude that the models that use the users' network embeddings alone significantly outperform the language model baselines in classifying fake news on the SD dataset. Moreover, our models that use unsupervised users' network embeddings can identify fake news on the SD dataset more accurately than the FANG models (GCN(G^{FANG}) and the GraphSage(G^{FANG})), which uses complex users' network embeddings that include additional relations such as the publisher network and the citation network, as well as the textual information from the tweets and news articles. Among the variants of our proposed UNES model, the variant using the GraphSAGE graph embeddings is the most effective. Thus, in response to the hypothesis (presented in Section 1.3) that user embeddings can be effective in identifying fake claims on Twitter, we conclude that unsupervised users embeddings can indeed assist the fake news detection on Twitter, and that the UNES variants GraphSAGE(G^{fr}) significantly outperform all other tested models and all the baselines, using all the training-testing split.

6.4.3 RQ 6.3: Followers or Friends?

To address RQ 6.3, we compare the experiment results between using a follower network and a friends network. Recall that users can have two types of relations with other users, namely through the following relation (Equation (6.3)) or through the friendship relation (Equation (6.4)), where the latter requires both users to follow each other. As discussed before, we instantiate the UNES model using user embeddings obtained from either the follower network (G^{fo}) or the friendship network (G^{fr}) for both the GraphSAGE and Cluster-GCN graph embedding models. Their respective performances are included in Table 6.4 and Figure 6.3. On analysing Table 6.4, we observe that with 90% of the data as training data, GraphSAGE(G^{fr}) outperforms GraphSAGE(G^{fo}) on all metrics, and Cluster-GCN(G^{fr}) outperforms Cluster-GCN(G^{fo}) on all metrics. Figure 6.3 shows that GraphSAGE(G^{fr}) outperforms GraphSAGE(G^{fo}) consistently on all metrics, regardless of the portion of training

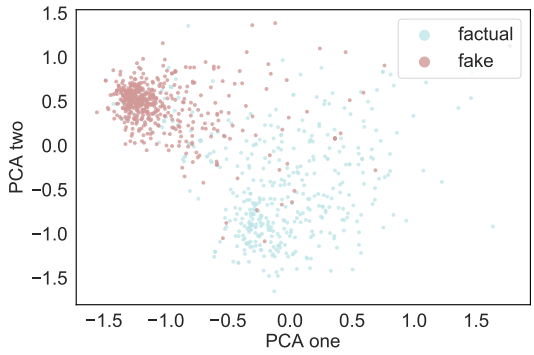


Figure 6.4: PCA mapping of the users engaged with 2 news articles related to immigration issues, on Friendship network. Red dots represent users who engaged with fake news "*Illegal Immigrant Deported 6 Times Charged in Felony Hit-and-Run of Family that Injured Children*", while teal dots represent users that engaged in factual news "*At Trump hotel site, immigrant workers wary*".

data used.

To understand this result, we analyse the statistical differences between the friendship network and the follower network in the SD dataset to investigate the dissimilarity of their embeddings. Recall Table 6.3, where it can be seen that, on average, users have a higher number of edges in the follower network than in the friendship network (1565.16 vs. 1388.37), echoing the fact that the friendship network on Twitter forms a more sparse network than the follower network. Although, intuitively speaking, the denser the network, the more information we can collect, we argue that the denser follower network may introduce more noise due to the lack of shared interests between followers and followees, compared to friends that share more interests and similar opinions, as represented by the friendship network.

Furthermore, the higher accuracy of the friend graph is advantageous from a data point of view: as noted in Section 6.3.1, we are limited in terms of possible Twitter API calls, which reduces the observable portion of the users' friends or follower networks. Indeed, from Table 6.3 it can be observed that the friendship network has a smaller proportion of users with more than 5000 friends (i.e., 10.64% of all news-engaging users in the SD dataset do not have all their friends downloaded); in contrast, the follower network has 6263 users with more than 5000 followers (17.50% of all news-engaging users).

Overall, in response to RQ 6.3 and **Limitation N2**, we conclude that in our experiments, with a limited number of Twitter API calls available, the users' friendship network is the most effective type of social network relationships, when used to construct the users' graph embeddings and for detecting fake news circulating on Twitter.

Table 6.5: Case study examples. The *Fake?* column indicates if the news article is fake news (Y) or not (N); # T_n describes the number of tweets are associated with the news article.

Case	Fake?	News article title	# T_n
UNES correct, BERT incorrect			
1	N	At Trump hotel site, immigrant workers wary	462
2	N	Charlie Hebdo : Le témoignage de la dessinatrice Coco.	132
3	N	Coldsip.com — News	155
4	Y	Illegal Immigrant Deported 6 Times Charged in Felony Hit-and-Run of Family that Injured Children.	588
5	Y	Sasha Obama Just Crashed Her Expensive New Car Into A Lake.	246
6	Y	Hundreds of people died after eating the Patti LaBelle brand Patti Sweet Potato Pie.	227
UNES incorrect, BERT correct			
7	N	UCLA Students protest after partygoers wear blackface at fraternity party.	246
8	N	Killer to face firing squad.	539
9	N	Caitlyn Jenner to receive courage award at ESPY's	797
10	Y	Obama Orders Chicago School to Let 'Transgender' Boy Use Girls' Locker Room.	398
11	Y	Peanut Butter and Jelly Deemed Racist.	835
12	Y	UPDATE: Second Roy Moore Accuser Works For Michelle Obama Right NOW.	3913

6.4.4 Case Study

In order to better demonstrate the successes and failures of our proposed UNES model, Table 6.5 presents a case study. Specifically, it shows that the UNES model makes correct predictions in cases 1-6, while the BERT model does not. Cases 7-12 shows the opposite examples, where the BERT model predicts correctly and UNES model does not. Table 6.5 also reports the number of tweets discussing each news article.

Indeed, we observe that our UNES model can classify the news correctly regardless of the language (case 2), or if the title is redacted (case 3) in cases 1-3. These cases show that our proposed model can be used universally across different languages, and when the news content is corrupted. Cases 4-6 in Table 6.5 show that the language used in the news title can mislead the language models into misclassifying the news as genuine, while our proposed UNES model can correctly classify the news as fake using the user embeddings from users engaged in such news.

On the other hand, cases 7-12 show difficult cases for the UNES model. These cases are mostly related to significant and controversial topics (such as racism, politician scandals, transgender right, immigrants), where people with different views can be easily drawn together and voice their opinions. We believe these cases are indeed difficult to identify with a user cluster-based model.

Furthermore, case 1 (factual) and case 4 (fake) are two news articles related to immigration issues in the US, that the UNES model correctly classified. Figure 6.4 shows the PCA projection of the embeddings of users engaged in these two cases. We observe two embedding clusters from Figure 6.4, one among users who commented on the fake news (case 4) and one among users who commented on the factual news (case 1). These two user clusters illustrate a clearer echo chamber effect on the immigration issue, in contracts to the lack of separation among the clusters observed in Figure 6.2(a). We believe these two clearly separated user clusters explain why UNES accurately classified these two cases. In order to further improve accuracy of identifying fake news, in the future we aim to combine UNES with language models to enhance the classification accuracy in such difficult examples.

6.5 Conclusions

In this chapter, we proposed a fake news detection model that leverages the network structure of Twitter. Our proposed model uses readily available friends/followers information of a set of users to build a social network structure. We first learnt user embeddings unsupervised from the user friendship/follower network, thus avoiding additional labelling requirements. We then used the user embeddings to classify news as fake or factual. Our proposed model UNES directly addresses **Gap 4**, where we stated that most fake news identification systems do not use user connections effectively.

We also demonstrated that a range of user network embedding methods, unsupervisedly trained on the users' relationships (i.e., followers, friends), can identify the user's clusters that engage with fake news, when tested on a large recent Twitter dataset. We observe a tighter cluster for the users who engaged in fake news than users who only engaged with factual news (shown in Figure 6.2) with $CEV = 0.8474$ for friendship network and $CEV = 0.8472$ for follower network, using PCA mapping. We conclude that the echo chamber effects are more pronounced in users engaging in fake news than in factual news.

When testing the effectiveness of user embeddings obtained using the readily available information (i.e., only the followers/friendship networks) on identifying fake news, we show that the UNES model can more accurately identify fake news than the existing SOTA models, as shown in Table 6.4 and Figure 6.3. Moreover, the current SOTA model GraphSage(G^{FANG}) uses both complex network information that requires additional labelling and text-based models. This finding addresses **Limitation N1**, as we show the readability available network features can be more information than the labour-intensive network features that need further labelling, and tackle the task of identifying fake news on Twitter more effectively.

Finally, we identified that the friendship network could more accurately help identify fake news than the follower network, within the limited access afforded to the Twitter social

networks given the limited Twitter API call rate. This observation is informative, because users tend to have fewer friends than followers (demonstrated in Table 6.3), thus helping us construct a smaller users' graph, and reduce computational cost. This finding addresses **Limitation N2**, because we show that the friendship network consistently outperforms the follower network in identifying fake news.

We note that our UNES model can best identify newly emerged fake news when users are within echo chambers, thus fake news published and spread by users that have few friends or followers might not be identified. In practical settings, we note the difficulty in training a large user network with limited resource, thus efficiency focused network embedding models maybe more practical in production.

Hence, in conclusion to the hypothesis presented in (Section 1.3), we conclude that unsupervised user network embedding can help identify echo chamber effects among Twitter users, and more accurately identify fake news than the SOTA model that uses complex network information. Thus, the main contributions of this chapter are as follows:

- We proposed to incorporate the idea of the echo chamber effect into the automatic fake news detection task. Specifically, we showed that training user network embeddings without prior knowledge of users' engagements with fake news can help identify user communities that frequently engage in and spread fake news and facilitate fake news detection in social media.
- We proposed a User Network Embedding Structure (UNES) model, which performs fake news classification on Twitter through the use of graph embeddings to represent Twitter users' social network structure. Compared to the approach of Nguyen et al. [128], UNES does **not** require any pre-annotated data (e.g., user type (individual users or publishers), users stance, and if they have engaged with fake news before).
- We observed that the user embeddings generated by UNES exhibit a clustering effect between users who engage with fake news and users who solely engage with factual news, despite not knowing if the users have engaged with fake news before.
- We also showed that using the social network's user connections alone to build network embeddings and using only users who engaged with the news when representing such news can significantly outperform the existing state-of-the-art fake news detection approaches that use both textual and complex social network features.

In the next chapter, we conduct an end-to-end study to demonstrate the effectiveness of our entire framework, and show that using the entire framework is more effective than using the individual models.

Chapter 7

End-to-End Evaluation

7.1 Introduction

In the previous chapters, we presented individual components of our proposed framework, FNDF. Specifically, Chapter 4 discussed how to identify check-worthy tweets/claims, and presented the experimental results using a range of language models and entity representation methods. We concluded that concatenating entity embeddings obtained using a KG embedding model, with language representations obtained using a language model, can most accurately identify check-worthy sentences and tweets. Chapter 5 discussed the task of identifying recurring fake news using an existing fake news dataset. We conducted experiments that combined a range of language models with the BM25 model to identify the recurring fake news. We concluded that the BM25 scores could indeed enhance the BERT language models in identifying the statements and tweets that *agree* with the existing fake news. Chapter 6 presented the task of using social network connections to identify fake news on the Twitter platform. We combined the embedded user representations of the engaging Twitter users to represent news, and detect the news that contains non-factual information. Our detailed experiments showed that our proposed user network embedding model UNES can more accurately identify fake news on Twitter than language models and is more accurate than sophisticated network models.

Recall the thesis statement presented in Section 1.3, where we hypothesised that by combining all three components, our proposed framework FNDF could effectively identify fake news circulating on Twitter in an end-to-end fashion. This chapter aims to test this hypothesis by conducting experiments using all three components according to the proposed framework presented in Section 3.2. At the same time, such end-to-end experiments allow us to address **Gap 5** identified in Section 2.4.4, which states that existing end-to-end fake news detection systems largely did not consider the filtering process, when not all tweets/claims/sentences require fact-checking; indeed, they generally overlooked the recurring fake news detection

process, and mostly did not apply dynamic user network information. Together, **Gaps 1-4** have been addressed.

In addition to the end-to-end evaluation, we also conduct an ablation study and report experimental results on the framework's performance if we omit any of the three components, to study the effectiveness of the individual components in our framework. We also conduct the experiments using two different datasets that focus on two different topics with different engaging users, thereby permitting to demonstrate the robustness of our framework to datasets involving unseen users.

This chapter is structured as follows: Section 7.2 presents the research questions we aim to answer in this chapter, and describes the methodologies of conducting the experiments that answer these research questions; Section 7.3 describes the experimental setup; Section 7.4 presents the experimental results and analysis; Concluding remarks follow in Section 7.5.

7.2 Methodology

We aim to address three research questions as follows:

- **RQ 7.1:** How does the Phase 1 Task 1 model, the check-worthiness ranking model, affect the FNDF's performance?
- **RQ 7.2:** What are the performance differences between the two tasks (Phase 2 Task 2 and Phase 2 Task 3) in the fact-checking phase of FNDF when detecting fake news?
- **RQ 7.3:** How robust is FNDF to news from an additional dataset that involves previously unseen users?

In the following, we describe the experimental datasets, and the construction of existing fake news collection and user networks in Section 7.2.1; the models we use in the end-to-end experiments are described in Section 7.2.2; Section 7.2.3 describes the workflow of our framework; the experimental designs are presented in Section 7.2.4.

7.2.1 Datasets Construction

In this chapter, we use three types of input data, namely, the experimental data, the existing fake news collection, and the user networks. We describe these datasets as follows:

7.2.1.1 Experimental Datasets

To study the effectiveness of our end-to-end framework, we use two experimental datasets as follows:

- We use the **MM-COVID** dataset provided by Li et.al. [96], which consists of source contents (news and tweets) related to COVID-19 that are labelled as fake or not, and their related tweets. Please refer to Table 5.4 in Section 5.3.1 for detailed statistics for this dataset.
- We use the **stance detection (SD)** dataset provided by Nguyen et al. [128][1]. This is the same dataset used in Chapter 6. Please refer to Table 6.2 in Section 6.3.1 for detailed statistics for the dataset.

For the MM-COVID dataset, we use the same training, validation, and test set split as presented in Section 5.3.1. For the SD dataset, we use the author provided 70% training, 15% evaluation, and 15% testing splits.

7.2.1.2 Existing Fake News Collection

We construct an **existing fake news collection** that contains previously identified fake news. The experimental datasets are compared against this collection to identify recurring fake news using the Phase 2 Task 2 model. We construct the existing fake news collection by combining all of the claims, tweets, and news titles labelled as fake news from the following published datasets:

1. *FAKENEWSNET* [162] contains gossip and political fake news gathered from *PolitiFact*[2], and *GossipCop*[3]. The dataset contains news articles labelled as fake or real. We collect the news titles from the fake news in the *FAKENEWSNET* as debunked fake news in our existing fake news collection.
2. *LIAR* [188] also contains claims gathered from *PolitiFact*. The Liar dataset differs from the *FAKENEWSNET* dataset because they did not collect the news stories being judged as fake or real. Rather, *LIAR* gathered the statements made in political speeches and debates. The statements are labelled as "pants-fire", "false", "barely true", "half-true", "mostly-true", and "true". In our existing fake news collection, we include statements labelled as "pants-fire", "false", and "barely true" as debunked fake news[4].

[1] https://github.com/nguyenvanhoang7398/FANG
[2] https://www.politifact.com/
[3] https://www.gossipcop.com/
[4] We acknowledge that by treating fake news as only a binary classification task we lose information, but multi-class fine grained fake news identification is out of scope for this thesis.

3. *PHEME* [210] contains tweets and their replies/retweets of five breaking news and four specific known rumours. We include all the 2,695 rumourous source tweets in our existing fake news collection.

4. *CoAID* [32]. As described in Section 5.3.1, the CoAID dataset contains claims, news articles, and engaged tweets that are labelled as fake and not fake. We include the titles of debunked fake news, fake claims and fake tweets in our existing fake news collection, which amount to 498 debunked fake news.

In total, we collected 13,891 claims/tweets/news titles that are labelled as fake.

7.2.1.3 User Friendship Networks

We construct a user friendship network for each experimental dataset (the MM-COVID dataset and the SD dataset). We use the same friendship network for the SD dataset as described in Section 6.3.1 – please refer to Table 6.3 for the statistics of the SD user friendship network. For the MM-COVID dataset, we follow the user network construction procedure laid out in Section 6.3.1. First, we download all the available tweets and their authors' **friends**. The number of downloaded friends is limited to a maximum of 5000 (the maximum number we can download as per the Twitter API limit) for each engaged user. Then, we remove the downloaded friends with less than five edges in the friend graph to reduce the size of the graph, which is able to be trained on a single RTX 3090 GPU with 24 GB memory.

In addition to the user friendship networks for each dataset, we also construct 2 *incremental* user friendship networks from the MM-COVID user friendship network and the SD user friendship network. That is, for the incremental MM-COVID user friendship network, only the engaged users of the SD data are added to the existing MM-COVID user friendship network, and the other way around. Finally, we create a *combined* user friendship network, where engaging users and their friends from both networks are used together in constructing the combined user friendship network. These three user friendship networks are constructed to study the robustness of our proposed framework, FNDF, in classifying news mostly discussed by unseen users. Thus, we list the following five user friendship networks we use in our evaluation:

- **MM-COVID user friendship network**: The user friendship network built with users present in the MM-COVID dataset and their friends.

- **SD user friendship network**: The user friendship network built with users present in the SD dataset and their friends.

Table 7.1: Statistics of the **engaged** users (EU) in the friendship networks, in five constructed user networks. EU denotes engaged users.

User network	# total nodes of G^{fr}	# total edges of G^{fr}	% of EU in MM-COVID connected	% of EU in SD connected
MM-COVID	58176	54,968,634	99.98	-
SD	31725	34,046,038	-	99.87
MM-COVID incremental	90169	65,712,042	99.98	8.6%
SD incremental	90169	34,052,734	1.6%	99.87
Combined	90169	80,342,653	99.98%	99.87%

- **Incremental MM-COVID user friendship network**: The user friendship network built with users presented in the MM-COVID dataset and their friends. Engaging users present in the SD dataset are added to the MM-COVID user friendship network, while the edges between the SD users and the existing nodes in the MM-COVID users' network are added incrementally, depending on if the SD engaging users are friends with the MM-COVID engaging users.
- **Incremental SD user friendship network**: This network is similar to the Incremental MM-COVID user friendship network, but the MM-COVID users are added to the SD user friendship network, and connections between the engaging MM-COVID users and engaging users in the SD user friendship network are added incrementally.
- **Combined user friendship network**: Finally, we construct a complete user friendship networks with both the SD dataset users and the MM-COVID dataset users, along with all their friends.

Table 7.1 presents the statistical information of the above mentioned 5 user friendship networks. The small number of engaged users in the MM-COVID dataset connected to the users in the SD dataset (1.6%), and vice versa (8.6%), indicates that the two datasets (MM-COVID and SD) have very different sets of engaging users. Note that some engaging users (0.02% and 0.13% in the MM-COVID dataset and the SD dataset respectively) have restricted access to their accounts, deleted their accounts, or have been suspended from Twitter, thus we can not access their accounts and collect their friendship lists.

7.2.2 Component Models

We describe the three models proposed in this thesis, presented in Chapters 4 - 6, and used in our end-to-end framework as follows:

1. **Task 1 model (denoted as T1)**: Identified as the most effective model in Chapter 4, we use the ALBERT language model to represent sentences and the BERTweet model to represent tweets, and use the ComplEx model to represent the entities identified. We concatenate two entity embeddings to represent an entity pair, and use the ranking model to rank all tweets and claims in the dataset. Finally, we retain 50% of the top-ranked claims and tweets as *check-worthy* and pass them on to the next models. We choose to pass 50% of the top-ranked claims and tweets to maximise Recall and minimise the additional tweets and claims going through T2 and T3, because there are 42.50% fake news in the SD dataset and 31.91% fake news in the MM-COVID dataset.

2. **Task 2 model (denoted as T2)**: Identified as the most effective model in Chapter 5, we use the combination of language representations of sentences and tweets with their BM25 scores to identify the recurring fake news. We use the BERTweet model to represent check-worthy tweets obtained from T1 and use the ALBERT language model (instead of the BERT model used in Chapter 5) to represent check-worthy sentences and claims obtained from T1. The sentences/claims and tweets identified as recurring fake news are assigned a final label as fake news, while the sentences/claims and tweets (along with the engaging users' ids and the user friendship network embeddings) that are not identified as recurring fake news are passed on to the next model.

3. **Task 3 model (denoted as T3)**: Identified as the most effective model in Chapter 6, we use the UNES model with user embeddings obtained using the GraphSage model to represent users that engaged with the check-worthy sentences/claims and tweets. We use the attention model [184], as described in Section 6.2.2, to obtain the final representation for the check-worthy sentences/claims and tweets using the user embeddings. Finally, we apply a dense layer to classify if the check-worthy sentences/claims and tweets are fake or not.

7.2.3 Framework Workflow

Building upon Figure 3.1 in Section 3.2, Figure 7.1 illustrates a simplified version of the workflow for our proposed framework, FNDF. Specifically, a group of tweets/sentences enter the framework as input data to T1 – the check-worthiness ranking model. T1 ranks the tweets/sentences according to their check-worthiness, and the sentences and tweets deemed check-worthy will then be sent to T2 – the recurring fake news detection model. T2 compares the check-worthy sentences/tweets against the existing fake news collection, to identify any recurring fake news. The check-worthy sentences and tweets identified as recurring fake news are directly labelled as fake news as their final label, while those non-recurring fake

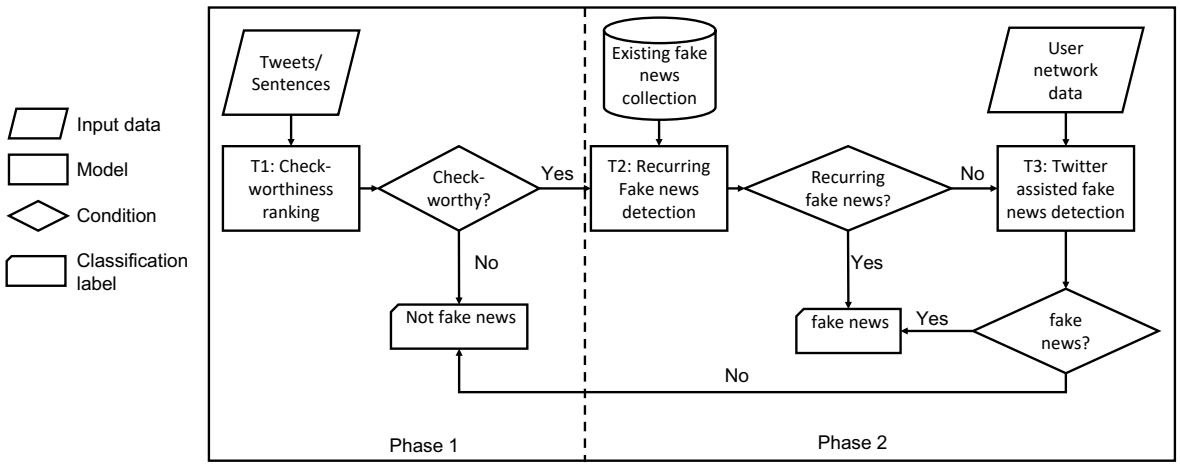


Figure 7.1: An illustration of the workflow of our proposed framework, FNDF. Phase 1 is the Check-Worthiness Detection Phase. Phase 2 is the Fact-Checking Phase. Task 1 in Phase 1 aims to rank tweets and sentences based on their check-worthiness, Task 2 in Phase 2 is dedicated to identifying recurring fake news, Task 3 in Phase 2 focuses on using the Twitter network in identifying fake news.

news will enter T3 as input data, along with the user friendship network, to be classified as fake news or not.

7.2.4 Experimental Designs

In order to study the role each component plays in our end-to-end framework, FNDF, and answer RQs 7.1 and 7.2, we perform an ablation study on the framework. We remove one component of the framework at a time, resulting in three variations of the FNDF using two components (i.e., T1 + T2, T1 + T3, T2 + T3). We also compare the results from using only one of the Phase 2 task models at a time (i.e., T1, T2), to observe the performance differences between these two task models.

To study the robustness of our proposed framework, FNDF, and answer RQ 7.3, we further conduct experiments that apply the five user networks presented in Section 7.2.1.3 to the two experimental datasets, and observe the performance differences from using these five different user networks.

7.3 Experimental Setup

In the following, we describe the approaches we use to represent both tweets and news articles, the user network embedding models, and the various baseline approaches and evaluation metrics we use in reporting our results.

7.3.1 Semantic Representations

In Chapter 4, we identified BERTweet as the most effective language model to represent tweets, while ALBERT is the best language model to represent sentences. Thus, in this chapter, we use BERTweet as the tweet embedding model and ALBERT as the sentence embedding model. We maintain the training settings for ALBERT and BERTweet as mentioned in Section 4.3.2.4. Specifically, we use the HuggingFace language model implementations [191][5]. We use the ALBERT-base-v2 English model (12-layer, 128-hidden, 12-heads, 1M parameters); and the BERTweet-base model (12-layer, 768-hidden, 12-heads, 135M parameters). We fine-tune ALBERT and BERTweet on the training datasets. All other hyper-parameters remain at their recommended settings.

7.3.2 Entity Representations

In this section, we describe the methods we use to identify named entities from text, and the entity embedding model we use to represent entities as vectors.

Named entity linking: To address the entities that occur in each sentence explicitly, we deploy a named entity linking method to extract entities from each sentence. In our experiments, we use DBpedia Spotlight[6] to extract entities from each sentence, with the confidence threshold set to 0.35. This setting is identical to the setting we used in Section 4.3.2.

Entity embedding model: We use ComplEx to represent entities as entity vectors, as it was identified as the most effective entity embedding method in Chapter 4. We use the same ComplEx model as in Section 4.3.2.3, where the ComplEx model is trained with triplets extracted from Freebase (FB15K) [16], using the code provided by Zheng et al. [205][7].

7.3.3 User Network Embedding Methods

As mentioned in Chapter 6, using the GraphSAGE [68] model on the user friendship network yields the best performance in identifying fake news on Twitter. Thus, we only use the GraphSage model to train the embedded user networks in this chapter. Specifically, we deploy the GraphSage model on the five user friendship networks described in Section 7.2.1.3. In particular, we train a GraphSAGE model for 30 epochs on each user friendship network, with 16 hidden units and 2 layers. The resulting models use a 100 dimension vector to represent each user. All the hyper-parameters we use in this chapter are the same as described in

[5] https://github.com/huggingface/transformers
[6] https://www.dbpedia-spotlight.org/
[7] https://github.com/awslabs/dgl-ke

Table 7.2: Performance comparison on the MM-COVID dataset using different framework variations. The user network model used in T3 is trained on the MM-COVID users' networks. Numbers in the significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.05$).

#	Model	Accuracy	P	R	F1	Significance
Individual component framework variations						
1	T2	0.9396	0.8824	0.9488	0.9144	2,7,8
2	T3	0.8669	0.8024	0.8072	0.8048	7
Two components framework variations						
3	T1 + T2	0.9406	0.9344	0.9331	0.9337	1,2,3,7,8
4	T1 + T3	0.8792	0.8794	0.7831	0.8150	7
5	T2 + T3	0.9191	0.8286	**0.9608**	0.8898	1-4,7,8
End-to-end framework						
6	End-to-end	**0.9468**	**0.9364**	0.9473	**0.9414**	1-4,7,8
Baselines						
7	Random	0.6602	0.4358	0.6602	0.5251	-
8	dEFEND	0.9103	0.9024	0.9072	0.9048	8, 2, 4

Section 6.3.3. Note that we represent the users who do not have any connections with other users in the network using an embedding vector of [-1, ..., -1].

7.3.4 Baselines

We report the performance of a random classifier using the stratified strategy for both the SD dataset and the MM-COVID dataset. We also report the performance of the current state-of-the-art models proposed by Nguyen et al. [128], namely the FANG model, for the SD dataset. We report the performance of the social context-based model dEFEND [161] as the baseline model for MM-COVID, following the MM-COVID paper [96]. The dEFEND model uses the user's reply sequences for fake news detection.

7.3.5 Evaluation Metrics

We report Precision, Recall, and F1 on the fake news class, and macro Accuracy as evaluation metrics.

7.4 Results and Analysis

In this section, we present the results of the experiments that address RQs 7.1-7.3. In particular, Tables 7.2 and 7.3 present the experimental results of using the FNDF variations on the MM-COVID dataset and the SD dataset, respectively. Tables 7.4 and 7.5 present the results

Table 7.3: Performance comparison on the SD dataset using different framework variations. The user network model used in T3 is trained on the SD users' networks. Numbers in the significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.05$).

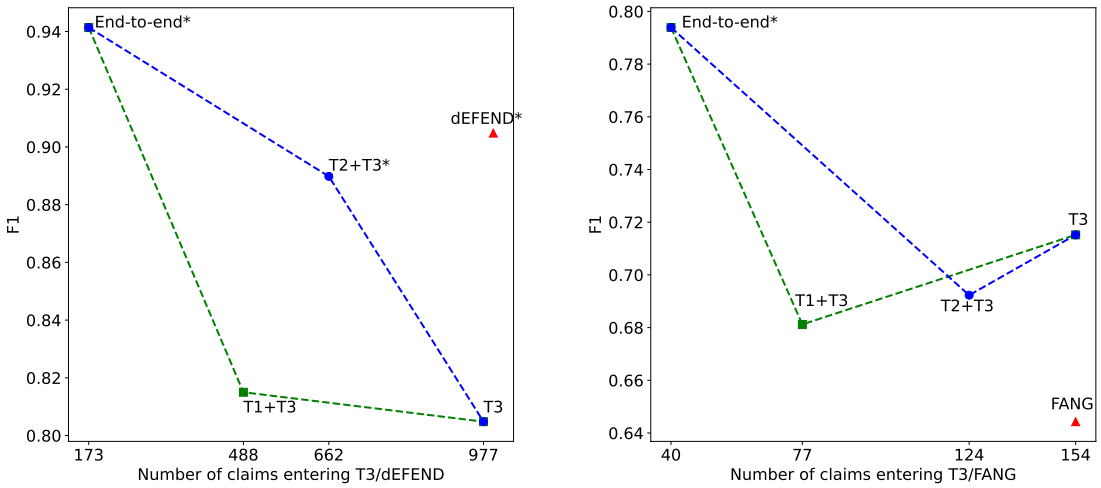
#	Model	Accuracy	P	R	F1	Significance
Individual component framework variations						
1	T2	0.6948	0.6852	0.5522	0.6116	2, 7
2	T3	0.6558	0.6429	**0.8060**	0.7152	2, 7, 8
Two components framework variations						
3	T1 + T2	0.7143	0.7447	0.5224	0.6140	1, 2, 7
4	T1 + T3	0.7143	0.6620	0.7015	0.6812	2, 7, 8
5	T2 + T3	0.6883	0.6067	**0.8060**	0.6923	1-4, 7, 8
End-to-end framework						
6	End-to-end	**0.8247**	**0.8125**	0.7761	**0.7939**	1-5 7, 8
Baselines						
7	Random	0.4351	0.3192	0.5649	0.4079	-
8	FANG	0.6689	0.6392	0.6495	0.6443	1,3

for the robustness of our proposed framework, FNDF, on the MM-COVID dataset and the SD dataset, respectively. Figure 7.2 presents the number of tweets/sentences entering T3 in framework variations versus the F1 score, on the MM-COVID and the SD datasets.

7.4.1 RQ 7.1: The Effectiveness of the Check-Worthy Ranking Phase

First, we evaluate the effectiveness of the Phase 1 Task 1 model. Tables 7.2 and 7.3 show the classification results of several framework variations and the end-to-end framework, on the MM-COVID and SD datasets, respectively. Note that we can not include the experimental results of using T1 alone in Tables 7.2 or 7.3, because T1 aims to filter a large number of sentences/claims and tweets, and has to be combined with either T2 or T3 to be able to detect fake news. Figure 7.2 shows the numbers of claims entering T3 for each variant, and their corresponding F1 scores.

Table 7.2 shows that all the tested models outperform the random baseline. T1 + T2 (row 3) and T1 + T3 (row 4) both outperform their single fact-checking tasks (T2 alone, row 1; and T3 alone, row 2) counterparts. In particular, we observe that T1 + T2 (row 3) significantly improves T1's (row 1) performance on Accuracy (0.9406 vs. 0.9396), Precision (0.9344 vs. 0.8824), and F1 (0.9337 vs. 0.9144), but shows a slight decrease in Recall (0.9331 vs. 0.9488). Similarly, T1 + T3 (row 4) also significantly outperforms T3 alone, on the accuracy, Precision, and F1 metrics. Table 7.2 shows that the end-to-end framework (row 6) outperforms T2 + T3 (row 5), and achieves the best performance on the accuracy, Precision, and



(a) The F1 score and the number of claims as input in T3/dEFEND on the MM-COVID dataset.

(b) The F1 score and the number of claims as input in T3/FANG on the SD dataset.

Figure 7.2: Figures of the number of claims as input in T3 in framework variants versus the F1 score, and their corresponding baseline on the MM-COVID dataset and the SD dataset, respectively. * denotes that the variation/baseline model significantly outperforms T3 alone.

F1 scores, among all the tested framework variations.

When tested on the SD dataset, we observe similar trends from Table 7.3. That is, adding T1 to T2, T1 to T3, or T1 to T2 + T3 helps improve their respective Accuracy, Precision, and F1 scores, while also hurting the Recall.

We postulate that since the filtering model (T1) removed 50% of the input tweets and claims as non-check-worthy, our framework can identify non-fake news more accurately, resulting in significantly higher Precision scores. However, T1 also identified a small number of fake news as not check-worthy, causing lower performances in Recall.

Moreover, Figure 7.2(a) shows that including T1 not only reduces the number of claims and tweets entering T3, but also increases the F1 performance on the MM-COVID dataset. However, from Figure 7.2(b) we observe that including T1 alone (the T1+ T3 variant) does not improve the F1 score than using T3 alone, but the end-to-end framework nonetheless significantly outperforms both T3 and T1 + T3.

Thus, in response to RQ 7.1, we conclude that T1, the check-worthiness identification model, can successfully filter out uncheck-worthy statements, resulting in better Accuracy and Precision performance in identifying fake news, while only affecting Recall by 1.4% on the MM-COVID dataset (row 5 vs. 6 in Table 7.2) and by 3.7% on the SD dataset(row 5 vs. 6 in Table 7.3). T1 also helps reduce the number of claims using computational expansive T3, while improving the F1 scores for the fake news identification task.

7.4.2 RQ 7.2: The Effectiveness of Components in Phase 2 of FNDF

To study the effectiveness of the Phase 2 components in the proposed framework, FNDF, we compare the performances of the T2 and T3 models.

We first compare the effectiveness of T2 and T3 on the MM-COVID dataset. Table 7.2 shows that using T2 alone (row 1) significantly outperforms using T3 alone (row 2) on all metrics (i.e., Accuracy: 0.,9396 vs. 0.8792; Precision: 0.8824 vs. 0.7635; Recall: 0.9488 vs. 0.8072; and F1: 0.9144 vs. 0.8491) in detecting fake news. We observe similar results by comparing using T1 + T2 (row 3) with using T1 + T3 (row 4). That is, on all metrics, using T1 + T2 significantly outperforms using T1 + T3. However, when comparing T2 + T3 (row 5) against using either T2 or T3 alone (row 1 and row 2), we observe that T2 + T3 significantly outperform using either one of the models on the Recall metrics. That is, T2 + T3 (0.9608) outperforms both T2 (0.9488) and T3 (0.8072) on Accuracy, Precision, Recall, and F1. This is likely due to the higher Precision (row 2 vs. 1, row 4 vs. 3) T2 obtained in detecting recurring fake news, while T3 obtained higher recall in identifying fake news that is not present in the existing fake news collection, based on the engaged user embeddings. This indicates that T2 and T3 are complementary to each other, and are both important in our end-to-end framework, FNDF.

When evaluating using the SD dataset in Table 7.3, we observe similar results. That is, T2 (row 1) significantly outperforms T3 (row 2) on Accuracy (0.6948 vs. 0.6558), and Precision (0.6852 vs. 0.6429). However, different from the results on the MM-COVID dataset presented in Table 7.2, T3 outperforms T2 in terms of Recall (0.6964 vs. 0.5804) and F1 (0.7152 vs. 0.6116) on the SD dataset. We observe similar differences between T1 + T2 (row 3) and T1 + T3 (row 4). Moreover, T2 + T3 (row 5) outperforms using either model alone on all metrics. Similar to the results obtained on the MM-COVID dataset, these results indicate that T2 focuses on the precise classification of recurring fake news, when the check-worthy claims and tweets are similar to previously encountered fake news, while T3 can identify fake news that is not present in our existing fake news datasets.

Furthermore, we note that T2 consistently outperforms T3 in terms of F1 on the MM-COVID dataset but not on the SD dataset. We postulate that this is because of the differences in the topic scope of the two datasets. Specifically, COVID-19 related fake news datasets are extensive due to the relatively narrow focus and high popularity of the COVID-19 topic, which is the only focus of the MM-COVID dataset. However, the SD dataset contains a wide range of fake news that is difficult to include in the existing fake news datasets.

In addition, Figure 7.2(a) shows that combing T2 and T3 can reduce the number of claims and tweets entering T3, while also increase the F1 performance significantly on the MM-

Table 7.4: Performance comparison on the MM-COVID dataset using different user networks. Numbers in the significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.05$).

#	User network	Accuracy	P	R	F1	Significance
1	MM-COVID	0.9468	**0.9364**	0.9473	**0.9414**	2, 4
2	SD	0.9396	0.8824	0.9488	0.9144	-
3	MM-COVID incremental	0.9478	0.9003	0.9518	0.9253	1, 2, 4
4	SD incremental	0.9417	0.8873	0.9488	0.9170	-
5	Combined	**0.9519**	0.9083	**0.9548**	0.9310	2, 3, 4

Table 7.5: Performance comparison on the SD dataset using different user networks. Numbers in the significance column indicate that the model is significantly better than the numbered model (McNemar's Test, $p<0.05$).

#	User network	Accuracy	P	R	F1	Significance
1	MM-COVID	0.6948	0.6852	0.5522	0.6116	-
2	SD user	0.8247	0.8125	0.7761	0.7939	1,3
3	MM-COVID incremental	0.7338	0.7167	0.6418	0.6772	1
4	SD incremental	0.8442	0.8413	0.7910	0.8154	1-3
5	Combined	**0.8701**	**0.8730**	**0.8209**	**0.8462**	1-4

COVID dataset, than using T3 alone. However, similar to including T1 on the SD dataset, Figure 7.2(b) shows that including T2 on the SD dataset does not improve the F1 score of T3, but the end-to-end framework that combines all three models indeed significantly outperforms both T3 and T2 + T3.

Thus, in response to RQ 7.2, we conclude that T2 – which aims to identify recurring fake news – can identify fake news more precisely than T3; while T3 – the social network assisted fake news identification model – can identify newly emerged fake news that is not recorded in the existing fake news datasets, resulting in higher recall. Thus, T2 and T3 can complement each other and achieve better performances when used together than those provided by individual models.

7.4.3 RQ 7.3: Robustness of the framework FNDF

Finally, we investigate the robustness of the proposed framework, FNDF, by conducting end-to-end evaluation using 5 different user networks presented in Section 7.2.1.3.

Table 7.4 presents the experimental results on the MM-COVID dataset, while varying the training data for the user network embeddings. We first observe that user network trained with the SD user network information (row 2) performs the worst among all tested user network embeddings. Similarly, using the SD incremental user network (row 4) does not significantly improve the end-to-end framework's performance. The poor robustness of our

proposed framework, when applied to news that involves unseen users, can be explained by the limited overlapping between the MM-COVID and SD users. Moreover, Table 7.1 shows that the MM-COVID users are not heavily interconnected with of the SD users network, as only 1.6% of the MM-COVID users are connected in the SD incremental set, which suggests that most of the MM-COVID users are represented as [-1, ..., -1], rendering T3's performance similar to random. However, we do observe that the end-to-end framework, FNDF, which applies user embeddings trained with both the MM-COVID incremental user network (row 3) and the combined user network (row 5), outperforms all the other models. This suggests that a larger and more interconnected user network can provide better user embeddings, and is essential for a robust performance from our framework in detecting the fake news.

Moving on to the experimental results on the SD dataset, Table 7.5 shows that, similar to that of the MM-COVID dataset, the end-to-end framework using the MM-COVID user network alone (row 1) significantly underperforms using the SD user network (row 2) alone, indicating poor robustness of our framework. However, using the MM-COVID incremental user network (row 3) significantly outperforms using only the MM-COVID user network. We postulate that the 9% of the SD users connected to the MM-COVID network (observed in Table 7.1) may have helped T3 identify a small number of fake news among the community, in addition to the fake news identified by T2 that identifies recurring fake news. Similar to the MM-COVID dataset, using user embeddings trained with the SD incremental user network and the combined user network significantly improves the framework performance than using the SD user network alone, over all the metrics.

These results indicate that our end-to-end framework is not robust when being applied to a news dataset, whose engaging users are very different from the user network the end-to-end framework is trained on. However, the framework's performance can be improved by including the unseen users (and their friends) from the new dataset in the user network, rather than using only the users' friendship network built on the old dataset, as shown by the results in Tables 7.4 and 7.5.

Thus, in response to RQ 7.3 and **Gap 5**, we conclude that our proposed end-to-end framework, FNDF, is robust in identifying fake news in a new dataset, only when the user network embeddings are updated to include the new engaging users and their friends.

7.5 Conclusions

This chapter conducted an ablation study on the framework components to analyse the effectiveness of each task model in our proposed framework, FNDF, using two publicly available datasets. We also investigated the robustness of the framework by alternating the user networks available to the framework.

Specifically, results from the ablation study (see Tables 7.2 and 7.3) showed that the Phase 1 Task 1 model, which aims to filter out the non-check-worthy sentences and claims, can indeed increase the overall Accuracy, Precision, and F1 performances of the framework, as it filters out some claims that the fact-checking models might misclassify. However, T1 cannot identify all the check-worthy sentences and claims, leading to a slightly lower Recall for the overall framework. Moreover, Figure 7.2 shows that including the check-worthiness ranking model can reduce the number of claims/sentences/tweets needing fact-checking, while also improving the F1 score of the end-to-end framework. Thus, we conclude that T1 is an important component of the end-to-end framework, FNDF, because it can successfully filter out uncheck-worthy statements, and improve Accuracy and Precision, as well as the F1 scores of the framework.

Phase 2 contains two models: T2 identifies recurring fake news, and T3 identifies fake news using social network user connections. Our experimental results from Tables 7.2 and 7.3 showed that T2 can identify recurring fake news with high Precision by comparing sentence/tweets with the existing fake news collection, while T3 can effectively identify newly emerged fake news that is not collected in the existing fake news collection, leading to a higher Recall than T2. Moreover, combining T2 and T3 leads to better Accuracy and F1 scores on both datasets. Thus, we conclude that T2 and T3 complement each other in detecting fake news online, and are both important components in FNDF.

Finally, the experimental results (see Tables 7.4 and 7.5) on the proposed FNDF showed limited robustness when the users engaging in the news are not well connected in the existing user networks (see Table 7.1). However, combining the engaged users and their friends in the two datasets makes the user friendship network denser and more connected, thus leading to better performance on both datasets. Thus, in answering **Gap 5**, we concluded that to best use FNDF, we need to update the user network periodically, to allow better connections among users and better fake news detection performances.

In the thesis statement presented in Section 1.3, we hypothesised that by combining all three components, our proposed FNDF framework could effectively identify fake news in an end-to-end fashion. Based on the experiments of this chapter, we conclude that combining all three proposed components indeed helps us identify fake news effectively, as the end-to-end framework achieved an Accuracy and F1 score above 0.94 for the MM-COVID dataset (see Table 7.2) and above 0.79 for the SD dataset (see Table 7.3), outperforming the SOTA models FANG on the SD dataset, and the dENFEND model on the MM-COVID dataset. Furthermore, Tables 7.4 and 7.5 showed that our framework could be robust when applied to unseen users, given a large enough user network.

In the next chapter, we close this thesis by summarising the results and conclusions from each chapter and providing possible new research directions uncovered by this work.

Chapter 8

Conclusions

In this chapter, Section 8.1 first provides the conclusions drawn from this thesis. Section 8.2 summarises the contributions of this thesis. We discuss possible future research directions for fake news detection online in Section 8.3. Finally, we present our closing remarks in Section 8.4.

8.1 Conclusions

In this thesis, we addressed the challenge of identifying fake news online with our proposed Fake News Detection Framework consisting of two phases and three tasks, namely, (1) Phase 1 Task 1, which combines embedded entities with language models to identify tweets and sentences that require fact-checking; (2) Phase 2 Task 2 uses an existing fake news collection for effective recurring fake news detection; (3) Phase 2 Task 3 leverages unsupervised Twitter users' network connections for identifying fake news.

In particular, in Chapter 4 (concerning Phase 1 Task 1), we proposed to capture the entity information and semantic information of a sentence/tweet/claim, by concatenating the embedded entity pair with its language model representation, for identifying check-worthy sentences/tweets. We observed that the ALBERT + ComplEx model – which uses the ALBERT model for sentence embeddings and the ComplEx model for entity embeddings – outperforms all other tested KG embedding models and language models combinations in both the sentences check-worthy ranking and the classification task. Similarly, the BERTweet + ComplEx model – which uses BERTweet for tweet embeddings and ComplEx for entity embeddings – outperforms all other tested KG embedding models and language models in both the tweets check-worthy ranking task and the classification task.

Chapter 5 (w.r.t. Phase 2 Task 2) proposed an ensemble model to classify the entailment of a check-worthy claim against existing debunked fake news in identifying recurring fake news

online. We showed that the classification model using the BERT language model could be enhanced with simple BM25 scores, classifying whether the two pieces of text agree with each other, using the WSDM 2019 Cup Fake News Challenge dataset. When tested on the MM-COVID dataset, the ensemble model of the BM25 scores and the BERT model can identify recurring fake news significantly more accurately than using the language model or the BM25 scores alone.

Chapter 6 (Phase 2 Task 3) proposed the UNES model that can effectively classify whether a tweet/news is fake based on its Twitter engagement. Specifically, UNES represents a check-worthy tweet/news as a vector, using the embedded entities of Twitter users who engaged with the tweet/news/claim. Our experiments showed that the UNES model outperforms many language models and complex network models that require handcrafted features.

Finally, in Chapter 7, we integrated all the proposed models to build the proposed end-to-end fake news detection framework. Our results show that our proposed framework, FNDF, can effectively identify fake news on two datasets, and demonstrate reasonable robustness when applied to the fake news dataset with unseen users engagement.

Next, we validate our thesis statement, proposed in Section 1.3, based on our empirical studies in Chapters 4 - 7. In summary, the key statement of this thesis is that effective fake news detection can be achieved in a two phases and three tasks framework (FNDF). Phase 1 Task 1 focuses on identifying check-worthy tweets and claims by enhancing language models with embedded entities. Phase 2 Task 2 identifies recurring fake news by an ensemble model of BM25 scores and language model representations to compare check-worthy claims and tweets with existing fake news. Phase 2 Task 3 focuses on identifying fake news with Twitter information by representing check-worthy tweets and news with user embeddings. Finally, our end-to-end framework can effectively identify fake news and be robust when applied to news involving previously unseen users. We present the hypotheses made in our thesis statement, and the evidence used to validate them, as follows:

- We hypothesised that by analysing entities in texts using an embedded knowledge graph, we could more accurately identify check-worthy claims from tweet content, articles, and debate quotes. We argue that we have validated this hypothesis in Chapter 4, where we showed that concatenating embedded entities with the pre-trained deep learning language model BERT can improve both the classification and the ranking tasks for identifying check-worthy tweets and sentences (see Tables 4.20 and 4.21). Furthermore, we also showed that among all the tested language models (i.e., TF.IDF, BiLSTM, BERT [40], ALBERT [89], RoBERTa [106], and BERTweet [126]), the ALBERT model performs the best at identifying check-worthy sentences, and that the BERTweet model performs the best at identifying check-worthy tweets. Among all the tested KG embedding models (i.e., Wikipedia2Vec [194], TransE [16], TransR [180],

RESCAL [100], DISTMult [129], and ComplEx [195]), Tables 4.21 and 4.20 showed that the ComplEx model performs the best in combination with the ALBERT model at identifying check-worthy sentences and the BERTweet model at identifying check-worthy tweets. Therefore, we have shown that enriching language models with embedded entity pair representations can indeed improve the language models' performances for identifying check-worthy tweets and sentences.

- We hypothesised that by comparing the targeted claim with an existing fake news collection, an ensemble model of a BM25 model and a deep neural network language model can accurately classify if a targeted check-worthy claim is highly similar to any existing fake news and thus is a resurfaced fake claim. Our experiments in Chapter 5 validated this hypothesis, with Table 5.5 showing our proposed ensemble model using the BM25 scores and the BERT language model representations can classify the entailment between two news titles more accurately than using either the BM25 scores or any pre-trained language models alone. Moreover, Table 5.7 showed that the ensemble model of BM25 and the BERT language model can effectively identify recurring fake news, by comparing tweets and claims that needed to be fact-checked with previously debunked fake news, compared to using either the BM25 scores or the BERT language model alone. Thus, we have shown that the BM25 scores can indeed enhance the language models in detecting the entailment among pairs of text, and thus improve the performance of language models in detecting recurring fake news.

- We hypothesised that user network embeddings trained with unlabelled user network data can identify the echo chamber effects among users and effectively identify fake claims on Twitter. We validated this hypothesis in Chapter 6, where Figure 6.2 showed that user embeddings trained with unlabelled user friendship networks have a CEV of 0.8474 between users who have engaged with fake news and users who have never engaged with fake news. This indicates that users who have and have never engaged with fake news can be distinguished, in the user friendship network. Furthermore, on a per topic basis, Figure 6.4 showed a more apparent separation among people who engaged with fake news and people who engaged with factual news w.r.t. the immigration issues in the US. Moreover, when testing the effectiveness of user embeddings obtained using the readily available information from the followers/friendship networks in identifying fake news, Table 6.4 and Figure 6.3 showed that our proposed UNES model can significantly (McNemar's Test, $p<0.01$) more accurately identify fake news than the existing SOTA model, which uses complex network including handcrafted features. Thus, we have shown that the unsupervised user embeddings learnt from Twitter users' friendship connections can indeed distinguish users who engage with fake news from users who have never engaged with fake news. We have also shown that using the engaged

users' embeddings to represent check-worthy tweets/sentences can identify fake news more effectively than using language models and Twitter networks with handcrafted features to represent check-worthy sentences/tweets.

- Finally, we hypothesised that our proposed framework, FNDF, which combines all the components, can effectively identify fake news circulating on Twitter end-to-end. We validated this claim in Chapter 7, where Tables 7.2 and 7.3 showed that our proposed FNDF outperforms the SOTA model FANG on the SD dataset, and the dEFEND model on the MM-COVID dataset. Moreover, combining the three tasks in our proposed framework FNDF achieved the highest accuracy in detecting fake news among all other framework variations with one or two components, using the MM-COVID and the SD datasets, indicating that the three tasks complement each other in our proposed framework FNDF. Furthermore, Tables 7.4 and 7.5 showed that our framework is robust even when applied to a new dataset involving unseen users, if we train the user embeddings on a combined network, where the user network involving unseen users is combined with the old user network.

Thus, all hypotheses have been validated, the thesis statement is thereby shown to have been upheld.

8.2 Contributions

The main contributions of this thesis are as follows:

- In Chapter 4, we represented tweets using language models that go beyond the bag of words and LSTM methods by leveraging the latest developments in deep neural language models (BERT [40], ALBERT [89], RoBERTa [106], and BERTweet [126]). We experiment with incorporating entity information within sentences and tweets, from the simple similarity and relatedness scores between the entities in a sentence to a more sophisticated entity representation obtained from KG embeddings. Our proposed model does not require the joint training of the language model and the entity representations, thereby providing greater flexibility for instantiating and deploying the model in fact-checking tasks.

- In Chapter 5, we compared a range of models in representing text, such as simple-embedding representations, BiLSTM and BERT, and identified the best practices in using BERT language model representations in classifying the relationship between Chinese news titles and between claims/tweets and debunked fake news. Our experiments showed that traditional BM25 retrieval scores can improve the performances

of deep neural network models in classifying whether a news title entails debunked fake news, such as the BiLSTM model and the BERT model. In answering our hypothesis that existing fake news collections can assist recurring fake news detection, we showed that using an ensemble model – which combines the BM25 scores and the BERT language model – in classifying the entailment relations between tweets/claims and previously debunked fake news can effectively identify recurring fake news from a set of check-worthy tweets and claims.

- Chapter 6 proposed to incorporate the idea of the echo chamber effect into the automatic fake news detection task. Specifically, we showed that training user network embeddings without prior knowledge of the users' engagements with fake news could help identify user communities that frequently engage in and spread fake news and thereby facilitating fake news detection in social media. We proposed a User Network Embedding Structure (UNES) model, which performs fake news classification on Twitter through graph embeddings to represent the Twitter users' social network structure. Compared to the approach of Nguyen et al. [128], UNES does **not** require any pre-annotated data, such as the user type (individual users or publishers), user's stance, and/or whether they have engaged with fake news before. We observed that the user embeddings generated by UNES exhibit a clustering effect between users who engage with fake news and users who solely engage with factual news, despite not knowing if the users have engaged with fake news before. We also showed that using the social network's user connections alone to build the network embeddings, and using only users who engaged with the news when representing such news, can significantly outperform an existing state-of-the-art fake news detection approach that uses both textual and complex social networks features.

- Chapter 7 conducted end-to-end experiments to investigate the effectiveness of our proposed framework, FNDF. IT also reported an ablation study to examine the effectiveness of each task model in identifying fake news, and conducted experiments to investigate the robustness of our proposed framework, FNDF. Specifically, FNDF outperformed the dEFEND model on the MM-COVID dataset by 4.0% and the FANG model by 23.2% in terms of F1 scores. Furthermore, regarding the individual task models, including the T1 – check-worthiness ranking model – in the end-to-end framework not only reduced the number of tweets and claims going to the check-worthy phase, but also resulted in 5.8% and 14.7% increases in terms of the F1 scores on the MM-COVID and the SD dataset, respectively. Similarly, including T2 –the recurring fake news identification model – in the end-to-end framework resulted in 15.5% and 16.5% increases in the F1 score on the MM-COVID and the SD dataset, respectively. Finally, including T3 – the user network assisted fake news detection model – in the

end-to-end framework resulted in a 0.8% and 29.3% increase of the F1 score on the MM-COVID and the SD dataset, respectively. Moreover, we observe that using the user network embedding trained on a combined user network of two datasets is on par with or outperforms the user network embedding trained for the single experimental dataset on the MM-COVID and the SD datasets, respectively, which indicates the robustness of our proposed framework, FNDF. Thus, we showed that the three task models are all important components of our end-to-end fake news detection framework, and that the FNDF is robust when applied to news involving unseen users, if the user friendship network embedding is updated with the unseen users and their friends.

8.3 Directions for Future Works

This section discusses possible directions for future research related to fake news detection. In particular, we discuss future research directions that have become apparent as a direct result of the work presented in this thesis.

- **Representing entities effectively in identifying check-worthy sentences and tweets:** In Chapter 4, we showed that an entity pair in a sentence can be represented by concatenating their respective embedded vectors together. Our experiments showed that this method can effectively represent entity pairs in sentences and tweets, and is effective in identifying check-worthy sentences and tweets when concatenated with language model embeddings. However, we limited the number of entities in each input instance to two and create more than one instance for each entity pair whether the sentence has more than two entities. In such cases, the current entity analysis setup becomes less than ideal. Thus, research on how to incorporate entities into language embedding representations, regardless of the number of entities that existed in the sentence/tweet, is an interesting future work that could be explored to support and facilitate check-worthy sentences/tweets identification and a range of other tasks (e.g., commonsense reasoning [38, 103], reasoning driven question answering [1, 21]) that can benefit from entity information.

- **Creating a large scale existing fake news collection:** In Chapter 5, we used an existing fake news collection to identify recurring fake news from check-worthy claims and tweets. In Chapter 7, we combined a range of publicly available fake news datasets [32, 162, 188, 210] to build an existing fake news collection, and compared a set of tweets and claims needing fact-checking to our constructed existing fake news collection. We showed that identifying recurring fake news can significantly improve the accuracy of identifying fake news in general. However, a systematic existing fake

news collection that includes all types of identified fake news retrospectively while updated periodically with newly emerged fake news does not exist. Thus, building such a collection is an important task in the research of fake news identification. In the future, we will explore providing an up-to-date large scale existing fake news collection to both journalists and researchers.

- **Integrating user descriptions into user network embeddings:** In Chapter 6, we proposed UNES, which uses user embeddings from unsupervised user network embeddings to represent news, and classify them as fake news or not. Our experiments showed significant improvements over the language model-based classifiers in identifying fake news on Twitter. However, the UNES model uses only the network structure of the Twitter network, which omits some important features that may help identify fake news, such as user descriptions [63], location information [80], or profile pictures [104]. In the future, we will explore how to best incorporate such user information into the network embedding models, so as to represent users more accurately, based on their description as well as their network connections on Twitter.

- **Combining textual features from the news with network features from the engaging users:** Figure 6.2 in Chapter 6 showed that the unsupervised user network embeddings can help us distinguish users who have commented on fake news from users who have never engaged in fake news, while Figure 6.4 showed a more apparent separation among the two groups of users on a single topic. In this case, identifying the topic of the check-worthy news and thus using the subset of the user embeddings related to the identified topic may provide more definitive and accurate classification results. Moreover, the textual analysis of the tweets and user embeddings may reduce the classification error rate in our proposed UNES model, especially when the check-worthy tweet/claim has sparked a large scale discussion from both the users who have engaged with fake news and users who never engage with fake news. Thus, in the future, we will research how to combine textual analysis models effectively with the user network-based fake news detection model for a more effective and accurate model for identifying fake news online.

- **Improving the Recall for the check-worthiness ranking model:** In Chapter 7 we conducted experiments to investigate the effectiveness of individual components of the framework. Tables 7.2 and 7.3 showed that including T1 – the check-worthiness ranking model – improved the F1 scores by 5.8% and 14.7% on the MM-COVID and the SD dataset, respectively. However, despite the improvement in the F1 scores, including T1 results in slight decreases in Recall for the end-to-end framework. It is important to retrieve as much fake news as possible in a fake news detection framework. Thus future works should focus on how to improve the Recall of the check-worthiness

ranking model.

- **Developing better strategy to leverage each model in our proposed framework:** Figure 7.1 in Section 7.2.3 demonstrated the workflow of our proposed framework. In Phase 2, T2 first predicts whether the check-worthy tweets/sentences/claims are recurring fake news. Then, if they are not identified as recurring fake news, T3 is used in predicting if they are fake news. Our results showed that this workflow can reduce the number of tweets going through the Twitter user network-assisted fake news detection model, UNES, while also improving the performance in identifying fake news. However, the simple if/then condition of deciding which model to use in detecting fake news could have been more intelligent. Thus, we recognise two strategies that could be investigated and developed in future studies: (1) one model that can achieve both identifying fake news as the T2 model and using the network assisted nature of the T3 model; (2) an ensemble model to combine the predictions of the T2 and T3 models.

8.4 Closing Remarks

In this thesis, we have addressed a challenging and important task, namely identifying fake news online. Effectively identifying fake news is an important task for several reasons. First, it is easy for anyone with Internet access to post any information online, with little to no consequences, which can spread worldwide very quickly. Secondly, there are not enough fact-checking websites and resources to fact-check all the online claims. Thirdly, users may encounter such false information, believe such information, and form small communities which reinforce false beliefs. Identifying fake news online is also a challenging task. For example, there is too much information created daily to fact-check all of them; fake news online can spread faster due to the timely fashion in which online data can be accessed worldwide.

We have shown that effectively identifying fake news online can be achieved by an end-to-end framework (FNDF), consisting of two phases and three tasks, which uses the language information from sentences/tweets, entities identified from the sentences/tweets/claims, and the Twitter users' engagement. Specifically, our comprehensive and extensive empirical experiments showed that our prosed framework, FNDF, can identify fake news more effectively than the SOTA models on two publicly available datasets. Moreover, we showed that the three task models are all important component in FNDF, as the end-to-end model significantly outperform and other framework variations, leading to more accurate fake news identification than using single models or two models combinations. Furthermore, we show that our proposed FNDF is robust when applied to fake news involving unseen users, by training user embeddings on a more extensive user network.

We have progressed in addressing some of the main gaps in identifying fake news online. However, there are still exciting aspects and complex challenges in the task of identification of fake news, which we highlighted in Section 8.3. In our discussions throughout this thesis, it has become apparent that pre-trained deep learning language models are effective in detecting fake news, while various other features, such as entities mentioned in the text, and users that engaged with the text, can be beneficial to tackling the task of fake news identification. We argue that pre-trained deep learning language models, and important features such as entities and users, will continue to be an essential trend in future research on online fake news detection.

Bibliography

[1] Aditya, S., Yang, Y., and Baral, C. (2018). Explicit reasoning over end-to-end neural architectures for visual question answering. In *Proceedings of the International Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference and AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 629–637.

[2] Alkhalifa, R., Yoong, T., Kochkina, E., Zubiaga, A., and Liakata, M. (2020). QMUL-SDS at CheckThat! 2020: Determining COVID-19 tweet check-worthiness using an enhanced CT-BERT with numeric expressions. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[3] Aloshban, N. (2020). ACT: Automatic fake news classification through self-attention. In *Proceedings of the ACM Conference on Web Science*, pages 115–124.

[4] Altun, B. and Kutlu, M. (2019). TOBB-ETU at CLEF 2019: Prioritizing claims based on check-worthiness. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[5] Apuke, O. D. and Omar, B. (2021). Fake news and COVID-19: Modelling the predictors of fake news sharing among social media users. *Telematics and Informatics*, 56:101475.

[6] Atanasova, P., Nakov, P., Karadzhov, G., Mohtarami, M., and Da San Martino, G. (2019a). Overview of the CLEF-2019 CheckThat! Lab on automatic identification and verification of claims. Task 1: Check-worthiness. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[7] Atanasova, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Karadzhov, G., Mihaylova, T., Mohtarami, M., and Glass, J. (2019b). Automatic fact-checking using context and discourse information. *Journal of Data and Information Quality (JDIQ)*, 11(3):1–27.

[8] Augenstein, I., Lioma, C., Wang, D., Chaves Lima, L., Hansen, C., Hansen, C., and Simonsen, J. G. (2019). MultiFC: A real-world multi-domain dataset for evidence-based

fact checking of claims. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 4685–4697.

[9] Austin, J. L. (1975). *How to do Things with Words*, volume 88. Oxford University Press.

[10] Balažević, I., Allen, C., and Hospedales, T. (2019). Multi-relational Poincaré graph embeddings. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 4463–4473.

[11] Barrón-Cedeno, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., et al. (2020). Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*, pages 215–236.

[12] Beltrán, J., Míguez, R., and Larraz, I. (2021). ClaimHunter: An unattended tool for automated claim detection on Twitter. In *Proceedings of the Wrokshop on Knowledge Graphs for Online Discourse Analysis in the ACM International Conference on World Wide Web*.

[13] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM International Conference on Management of Data*, pages 1247–1250.

[14] Bonfadelli, H. (2002). The internet and knowledge gaps: A theoretical and empirical investigation. *European Journal of Communication*, 17(1):65–84.

[15] Bordes, A., Glorot, X., Weston, J., and Bengio, Y. (2014). A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

[16] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 2787–2795.

[17] Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Çelikyilmaz, A., and Choi, Y. (2019). COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

[18] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

[19] Bratu, S. (2020). The fake news sociology of covid-19 pandemic fear: Dangerously inaccurate beliefs, emotional contagion, and conspiracy ideation. *Linguistic and Philosophical Investigations*, 19:128–135.

[20] Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Conference on Applied natural Language Processing*, pages 152–155.

[21] Cao, Q., Liang, X., Li, B., and Lin, L. (2019). Interpretable visual question answering by reasoning on dependency trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):887–901.

[22] Cerone, A., Naghizade, E., Scholer, F., Mallal, D., Skelton, R., and Spina, D. (2020). Watch'n'Check: Towards a social media monitoring tool to assist fact-checking experts. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics*, pages 607–613.

[23] Chami, I., Wolf, A., Juan, D.-C., Sala, F., Ravi, S., and Ré, C. (2020). Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914.

[24] Cheema, G. S., Hakimov, S., and Ewerth, R. (2020). Check square at CheckThat! 2020: Claim detection in social media via fusion of transformer and syntactic features. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[25] Chen, J., Zhu, J., and Song, L. (2018). Stochastic training of graph convolutional networks with variance reduction. In *Proceedings of the ACM International Conference on Machine Learning*, pages 942–950.

[26] Chen, S., Wang, J., Jiang, F., and Lin, C.-Y. (2020). Improving entity linking by modeling latent entityttype information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7529–7537.

[27] Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S., and Hsieh, C.-J. (2019). Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 257–266.

[28] Cho, J., Ahmed, S., Hilbert, M., Liu, B., and Luu, J. (2020). Do search algorithms endanger democracy? An experimental investigation of algorithm effects on political polarization. *Journal of Broadcasting & Electronic Media*, 64(2):150–172.

[29] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

[30] Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). Computational fact checking from knowledge networks. *PloS One*, 10(6):e0128193.

[31] Coca, L., Cusmuliuc, C.-G., and Iftene, A. (2019). CheckThat! 2019 UAICS. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[32] Cui, L. and Lee, D. (2020). CoAid: COVID-19 healthcare misinformation dataset. *arXiv preprint arXiv:2006.00885*.

[33] Cusmuliuc, C.-G., Coca, L.-G., and Iftene, A. (2020). UAICS at CheckThat! 2020: Fact-checking claim prioritization. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR WorkshopWorkshop*.

[34] Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., and Ali, Z. S. (2020). Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[35] Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Proceedings of the Machine Learning Challenges Workshop in Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

[36] Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the International Conference on Semantic Systems (I-Semantics)*, pages 121–124.

[37] Das, S. D., Basak, A., and Dutta, S. (2021). A heuristic-driven ensemble framework for COVID-19 fake news detection. In *Proceedings of the International Workshop on Combating Online Hostile Post in Regional Languages during Emergency Situation*, pages 164–176.

[38] Davis, E. and Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103.

[39] Dehghani, M., Zamani, H., Severyn, A., Kamps, J., and Croft, W. B. (2017). Neural ranking models with weak supervision. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 65–74.

[40] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the ACL Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

[41] Dhar, R., Dutta, S., and Das, D. (2019). A hybrid model to rank sentences for check-worthiness. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[42] Dib, F., Mayaud, P., Chauvin, P., and Launay, O. (2021). Online mis/disinformation and vaccine hesitancy in the era of COVID-19: Why we need an ehealth literacy revolution. *Human Vaccines and Immunotherapeutics*, pages 1–3.

[43] Dickerson, J. P., Kagan, V., and Subrahmanian, V. (2014). Using sentiment to detect bots on Twitter: Are humans more opinionated than bots? In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 620–627.

[44] Dictionary, M.-W. (2002). Merriam-webster. *On-Line at http://www. mw. com/home. htm*, 8.

[45] Du, X., Yan, J., and Zha, H. (2019). Joint link prediction and network alignment via cross-graph embedding. In *Proceedings of the ACM International Joint Conference on Artificial Intelligence*, pages 2251–2257.

[46] Fan, Y., Pang, L., Hou, J., Guo, J., and Lan, Y. (2019). MatchZoo: A toolkit for deep text matching. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 1297–1300.

[47] Fang, A., Macdonald, C., Ounis, I., and Habel, P. (2016a). Examining the cohernece of the topic ranked Tweets topics. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 825–828.

[48] Fang, A., Macdonald, C., Ounis, I., and Habel, P. (2016b). Topics in tweets: A user study of topic coherence metrics for Twitter data. In *Proceedings of the European Conference on Information Retrieval*, pages 492–504.

[49] Fang, A., Macdonald, C., Ounis, I., and Habel, P. (2016c). Using word embedding to evaluate the coherence of topics from Twitter data. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 1057–1060.

[50] Fang, Y., Gao, J., Huang, C., Peng, H., and Wu, R. (2019). Self multi-head attention-based convolutional neural networks for fake news detection. *PloS One*, 14(9):e0222713.

[51] Faragó, L., Kende, A., and Krekó, P. (2020). We only believe in news that we doctored ourselves. *Social Psychology*, 51(2):77–90.

[52] Favano, L., Carman, M., and Simonsen, J. G. (2019). TheEarthIsFlat's submission to CLEF'19 CheckThat! challenge. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[53] Feng, S., Banerjee, R., and Choi, Y. (2012). Syntactic stylometry for deception detection. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 171–175.

[54] Gasior, J. and Przybyła, P. (2019). The IPIPAN team participation in the check-worthiness task of the CLEF2019 CheckThat! Lab. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[55] Gelfert, A. (2018). Fake news: A definition. *Informal Logic*, 38(1):84–117.

[56] Gencheva, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., and Koychev, I. (2017). A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the ACL International Conference Recent Advances in Natural Language Processing*, pages 267–276.

[57] Ghenai, A. and Mejova, Y. (2017). Catching Zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on Twitter. In *Proceedings of the IEEE International Conference on Healthcare Informatics*, pages 518–518.

[58] Grant, M. (2004). *Greek and Roman Historians: Information and Misinformation*. Routledge.

[59] Graves, L. (2017). Anatomy of a fact check: Objective practice and the contested epistemology of fact checking. *Communication, Culture & Critique*, 10(3):518–537.

[60] Grech, V. (2017). Fake news and post-truth pronouncements in general and in early human development. *Early Human Development*, 115:118–120.

[61] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 855–864.

[62] Guo, Z. and Barbosa, D. (2014). Robust entity linking via random walks. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 499–508.

[63] Gupta, A., Joshi, A., and Kumaraguru, P. (2012). Identifying and characterizing user communities on Twitter during crisis events. In *Proceedings of the ACM Workshop on Data-Driven User Behavioral Modelling and Mining from Social Media*, pages 23–26.

[64] Gupta, A., Lamba, H., Kumaraguru, P., and Joshi, A. (2013). Faking Sandy: Characterizing and identifying fake images on Twitter during Hurricane Sandy. In *Proceedings of the ACM International Conference on World Wide Web*, pages 729–736.

[65] Halpern, D., Valenzuela, S., Katz, J., and Miranda, J. P. (2019). From belief in conspiracy theories to trust in others: Which factors influence exposure, believing and sharing fake news. In *Proceedings of the International Conference on Human-Computer Interaction*, pages 217–232.

[66] Hamidian, S. and Diab, M. (2015). Rumor detection and classification for Twitter data. In *Proceedings of the International Conference on Social Media Technologies, Communication, and Informatics*, pages 71–77.

[67] Hamidian, S. and Diab, M. (2016). Rumor identification and belief investigation on Twitter. In *Proceedings of the ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 3–8.

[68] Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 1025–1035.

[69] Hansen, C., Hansen, C., Simonsen, J., and Lioma, C. (2019). Neural weakly supervised fact check-worthiness detection with contrastive sampling-based ranking loss. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[70] Hassan, N., Arslan, F., Li, C., and Tremayne, M. (2017). Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812.

[71] Hassan, N., Li, C., and Tremayne, M. (2015). Detecting check-worthy factual claims in presidential debates. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 1835–1838.

[72] He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., and Wang, H. (2013). Learning entity representation for entity disambiguation. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 30–34.

[73] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[74] Jaradat, I., Gencheva, P., Barrón-Cedeño, A., Màrquez, L., and Nakov, P. (2018). ClaimRank: Detecting check-worthy claims in Arabic and Eglish. In *Proceedings of the ACL Conference of the North American Chapter of the Association for Computational Linguistics*, pages 26–30.

[75] Jin, Z., Cao, J., Zhang, Y., and Luo, J. (2016). News verification by exploiting conflicting social viewpoints in microblogs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2972–2978.

[76] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

[77] Karagiannis, G., Saeed, M., Papotti, P., and Trummer, I. (2020). Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification. In *Proceedings of the VLDB Endowment*, pages 2508–2521.

[78] Kartal, Y. S. and Kutlu, M. (2020). TOBB ETU at CheckThat! 2020: Prioritizing English and Arabic claims based on check-worthiness. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[79] Kasami, T. (1966). An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report*, R-257.

[80] Kazai, G., Yusof, I., and Clarke, D. (2016). Personalised news and blog recommendations based on user location, Facebook and Twitter user profiling. In *Proceedings of the ACM International conference on Research and Development in Information Retrieval*, pages 1129–1132.

[81] Kazemi, S. M. and Poole, D. (2018). SimplE embedding for link prediction in knowledge graphs. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 4289–4300.

[82] Kipf, T. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

[83] Kittur, A., Chi, E., Pendleton, B. A., Suh, B., and Mytkowicz, T. (2007). Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *World Wide Web*, 1(2):19–28.

[84] Kochkina, E., Liakata, M., and Augenstein, I. (2017). Turing at SemEval-2017 task 8: Sequential approach to rumour stance classification with Branch-LSTM. In *Proceedings of the ACL International Workshop on Semantic Evaluation*, pages 475–480.

[85] Kolluri, N. L. and Murthy, D. (2021). CoVerifi: A COVID-19 news verification system. *Online Social Networks and Media*, 22:100123.

[86] Kristiyono, J. and Jayanti, O. R. (2017/11). Fake news (hoax) and paranoid frame of min of social media user. In *Proceedings of the International Conference on Transformation in Communications*, pages 41–44.

[87] Kucharski, A. (2016). Post-truth: Study epidemiology of fake news. *Nature*, 540(7634):525.

[88] Ladd, J. M. (2013). The era of media distrust and its consequences for perceptions of political reality. *New Directions in Media and Politics*, pages 24–44.

[89] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the International Conference on Learning Representations*.

[90] Latkin, C., Dayton, L., Strickland, J. C., Colon, B., Rimal, R., and Boodram, B. (2020). An assessment of the rapid decline of trust in US sources of public information about COVID-19. *Journal of Health Communication*, 25(10):764–773.

[91] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the ACM International Conference on Machine Learning*, pages 1188–1196.

[92] Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the ACL Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 687–692.

[93] Lewandowsky, S. (2021). Climate change disinformation and how to combat it. *Annual Review of Public Health*, 42:1–21.

[94] Lewandowsky, S., Ecker, U., Seifert, C., Schwarz, N., and Cook, J. (2012). Misinformation and its correction: Continued influence and successful debiasing. *Psychological Science in the Public Interest*, 13(3):106–131.

[95] Li, D. and Madden, A. (2019). Cascade embedding model for knowledge graph inference and retrieval. *Information Processing & Management*, 56(6):102093.

[96] Li, Y., Jiang, B., Shu, K., and Liu, H. (2020). MM-COVID: A multilingual and multimodal data repository for combating COVID-19 disinformation. In *Proceedings of the IEEE International Conference on Big Data*, pages 4325–4330.

[97] Liang, G., He, W., Xu, C., Chen, L., and Zeng, J. (2015). Rumor identification in microblogging systems based on users' behavior. *IEEE Transactions on Computational Social Systems*, 2(3):99–108.

[98] Limba, T. and Šidlauskas, A. (2019). Peculiarities of anonymous comments' management: a case study of Lithuanian news portals. *Journal of Entrepreneurship and Sustainability Center*, 5(4):875–889.

[99] Lin, P., Song, Q., and Wu, Y. (2018). Fact checking in knowledge graphs with ontological subgraph patterns. *Data Science and Engineering*, 3(4):341–358.

[100] Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015a). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2181–2187.

[101] Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015b). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI International Conference on Artificial Intelligence*, pages 2181–2187.

[102] Ling, R. (2020). Confirmation bias in the era of mobile news consumption: the social and psychological dimensions. *Digital Journalism*, 8(5):596–604.

[103] Liu, H. and Singh, P. (2004). ConceptNet— a practical commonsense reasoning toolkit. *BT Technology Journal*, 22(4):211–226.

[104] Liu, L., Preotiuc-Pietro, D., Riahi, S. Z., E., M. M., and Ungar, L. (2016). Analyzing personality through social media profile picture choice. In *Proceedings of the AAAI International Conference on Web and Social Media*, pages 211–220.

[105] Liu, S., Liu, S., and Ren, L. (2019a). Trust or suspect? an empirical ensemble framework for fake news classification. In *Proceedings of the WSDM 2019 Cup Fake News Classification Challenge in the International Conference on Web Search and Data Mining*.

[106] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

[107] Lv, S., Guo, D., Xu, J., Tang, D., Duan, N., Gong, M., Shou, L., Jiang, D., Cao, G., and Hu, S. (2020). Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8449–8456.

[108] Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., and Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the ACM International Joint Conference on Artificial Intelligence*, pages 3818–3824.

[109] Ma, J., Gao, W., and Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 708–717.

[110] MacAvaney, S., Yates, A., Cohan, A., and Goharian, N. (2019). CEDR: Contextualized embeddings for document ranking. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 1101–1104.

[111] MacCartney, B. and Manning, C. D. (2008). Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the International Conference on Computational Linguistics*, pages 521–528.

[112] Majithia, S., Arslan, F., Lubal, S., Jimenez, D., Arora, P., Caraballo, J., and Li, C. (2019). ClaimPortal: Integrated monitoring, searching, checking, and analytics of factual claims on Twitter. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 153–158.

[113] Martinez-Rico, J., Martinez-Romo, J., and Araujo, L. (2021). NLP&IR@ uned at CheckThat! 2021: Check-worthiness estimation and fake news detection using transformer models. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[114] Martinez-Rico, J. R., Araujo, L., and Martinez-Romo, J. (2020). NLP&IR@ uned at CheckThat! 2020: A preliminary approach for check-worthiness and claim retrieval tasks using neural networks and graphs. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[115] McDonald, T., Dong, Z., Zhang, Y., Hampson, R., Young, J., Cao, Q., Leidner, J. L., and Stevenson, M. (2020). The University of Sheffield at CheckThat! 2020: Claim

identification and verification on Twitter. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[116] Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of the ACM International Conference on Semantic Systems*, pages 1–8.

[117] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

[118] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 3111–3119.

[119] Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

[120] Mitra, T. and Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings of the International Conference on Web and Social Media*, pages 258–267.

[121] Moreno, J. G., Besançon, R., Beaumont, R., D'hondt, E., Ligozat, A.-L., Rosset, S., Tannier, X., and Grau, B. (2017). Combining word and entity embeddings for entity linking. In *Proceedings of the European Semantic Web Conference*, pages 337–352.

[122] Mueller, J. and Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2786–2792.

[123] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the ACM International Conference on Machine Learning*, pages 807–814.

[124] Nakov, P., Da San Martino, G., Elsayed, T., Barrón-Cedeño, A., Míguez, R., Shaar, S., Alam, F., Haouari, F., Hasanain, M., Mansour, W., et al. (2021). Overview of the CLEF–2021 CheckThat! Lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[125] Nathani, D., Chauhan, J., Sharma, C., and Kaul, M. (2019). Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723.

[126] Nguyen, D. Q., Vu, T., and Nguyen, A. T. (2020a). BERTweet: A pre-trained language model for Eglish tweets. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

[127] Nguyen, T. T., Weidlich, M., Yin, H., Zheng, B., Nguyen, Q. H., and Nguyen, Q. V. H. (2020b). FactCatch: Incremental pay-as-you-go fact checking with minimal user effort. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 2165–2168.

[128] Nguyen, V.-H., Sugiyama, K., Nakov, P., and Kan, M.-Y. (2020c). Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 1165–1174.

[129] Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the ACM International Conference on Machine Learning*, pages 809–816.

[130] Nielsen, R. K., Fletcher, R., Newman, N., Brennen, J. S., and Howard, P. N. (2020). *Navigating the 'infodemic': How People in Six Countries Access and Rate News and Information about Coronavirus*. Reuters Institute.

[131] Nikolov, A., Da San Martino, G., Koychev, I., and Nakov, P. (2020). Team_Alex at CheckThat! 2020: Identifying check-worthy tweets with transformer models. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[132] Novak, V. (2020). Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[133] Oraby, S., Reed, L., Compton, R., Riloff, E., Walker, M., and Whittaker, S. (2015). And that's a fact: Distinguishing factual and emotional argumentation in online dialogue. In *Proceedings of the 2nd Workshop on Argumentation Mining in the ACL Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 116–126.

[134] Patwari, A., Goldwasser, D., and Bagchi, S. (2017). TATHYA: A multi-classifier system for detecting check-worthy statements in political debates. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 2259–2262.

[135] Pennycook, G., Cannon, T., and Rand, D. (2018). Prior exposure increases perceived accuracy of fake news. *Journal of Experimental Psychology: General*, 147(12):1865–1880.

[136] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 701–710.

[137] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the ACL Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2227–2237.

[138] Peters, M. E., Neumann, M., Logan IV, R. L., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge enhanced contextual word representations. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 43–54.

[139] Pham, L. (2019). Transferring, transforming, ensembling: the novel formula of identifying fake news. In *Proceedings of the WSDM 2019 Cup Fake News Classification Challenge in the International Conference on Web Search and Data Mining*.

[140] Pierson, P. and Schickler, E. (2020). Madison's constitution under stress: A developmental analysis of political polarization. *Annual Review of Political Science*, 23:37–58.

[141] Polage, D. (2012). Making up history: False memories of fake news stories. *Europe's Journal of Psychology*, 8(2):245–250.

[142] Prior, M. (2013). Media and political polarization. *Annual Review of Political Science*, 16:101–127.

[143] Ramachandran, G., Nemeth, D., Neville, D., Zhelezov, D., Yalçin, A., Fohrmann, O., and Krishnamachari, B. (2020). WhistleBlower: Towards a decentralized and open platform for spotting fake news. In *Proceedings of the IEEE International Conference on Blockchain*, pages 154–161.

[144] Rath, B., Salecha, A., and Srivastava, J. (2020). Detecting fake news spreaders in social networks using inductive representation learning. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 182–189.

[145] Reddy, H., Raj, N., Gala, M., and Basava, A. (2020). Text-mining-based fake news detection using ensemble methods. *International Journal of Automation and Computing*, 17(2):210–221.

[146] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at TREC-3. *NIST special publication*, (500225):109–123.

[147] Röchert, D., Shahi, G. K., Neubaum, G., Ross, B., and Stieglitz, S. (2021). The networked context of COVID-19 misinformation: Informational homogeneity on YouTube at the beginning of the pandemic. *Online Social Networks and Media*, 26:100164.

[148] Roets, A. et al. (2017). 'fake news': Incorrect, but hard to correct. the role of cognitive ability on the impact of false information on social impressions. *Intelligence*, 65:107–110.

[149] Rösner, L. and Krämer, N. C. (2016). Verbal renting in the social web: Effects of anonymity and group norms on aggressive language use in online comments. *Social Media + Society*, 2(3):2056305116664220.

[150] Rosnow, R. L. (1988). Rumor as communication: A contextualist approach. *Journal of Communication*, 38(1):12–28.

[151] Rossi, A., Barbosa, D., Firmani, D., Matinata, A., and Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2):1–49.

[152] Ruchansky, N., Seo, S., and Liu, Y. (2017). CSI: A hybrid deep model for fake news. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 797–806.

[153] Saeed, R., Afzal, H., Abbas, H., and Fatima, M. (2021). Enriching conventional ensemble learner with deep contextual semantics to detect fake news in Urdu. *ACM Transactions on Asian and Low-Resource Language Information Processing.*, 21(1).

[154] Santia, G. C. and Williams, J. R. (2018). Buzzface: A news veracity dataset with facebook user commentary and egos. In *Proceedings of the International Conference on Web and Social Media*, pages 531–540.

[155] Saward, M. (2006). The representative claim. *Contemporary Political Theory*, 5(3):297–318.

[156] Schlicht, I. B., de Paula, A. F. M., and Rosso, P. (2021). UPV at CheckThat! 2021: Mitigating cultural differences for identifying multilingual check-worthy claims. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[157] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[158] Shi, B. and Weninger, T. (2017). ProjE: Embedding projection for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1236–1242.

[159] Shin, J., Jian, L., Driscoll, K., and Bar, F. (2018). The diffusion of misinformation on social media: Temporal pattern, message, and source. *Computers in Human Behavior*, 83:278–287.

[160] Shin, J. and Thorson, K. (2017). Partisan selective sharing: The biased diffusion of fact-checking messages on social media. *Journal of Communication*, 67(2):233–255.

[161] Shu, K., Mahudeswaran, D., and Liu, H. (2019). FakeNewsTracker: A tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, 25(1):60–71.

[162] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

[163] Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM Knowledge Discovery and Data Mining: Explorations Newsletter*, 19(1):22–36.

[164] Singhal, A. et al. (2001). Modern information retrieval: A brief overview. *Bulletin of the Technical Committee on Data Engineering*, 24(4):35–44.

[165] Sosnkowski, A., Fung, C. J., and Ramkumar, S. (2021). An analysis of Twitter users' long term political view migration using cross-account data mining. *Online Social Networks and Media*, 26:100177.

[166] Su, T., Fang, A., McCreadie, R., Macdonald, C., and Ounis, I. (2018). On refining Twitter lists as ground truth data for multi-community user classification. In *Proceedings of the European Conference on Information Retrieval*, pages 765–772.

[167] Su, T., Macdonald, C., and Ounis, I. (2019a). Ensembles of recurrent networks for classifying the relationship of fake news titles. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 893–896.

[168] Su, T., Macdonald, C., and Ounis, I. (2019b). Entity detection for check-worthiness prediction: Glasgow Terrier at CLEF CheckThat! 2019. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages in CEUR Workshop*.

[169] Su, T., Macdonald, C., and Ounis, I. (2022a). Entity-assisted language models for identifying check-worthy sentences. *Under Review.*

[170] Su, T., Macdonald, C., and Ounis, I. (2022b). FNDF: an end-to-end fake news detection framework. In *Under Review.*

[171] Su, T., Macdonald, C., and Ounis, I. (2022c). Leveraging users' social network embeddings for fake news detection on Twitter. *Under Review.*

[172] Su, T., Wang, X., Macdonald, C., and Ounis, I. (2019c). University of Glasgow Terrier Team at the TREC 2019 Deep Learning Track. In *TREC*.

[173] Suiter, J. and Fletcher, R. (2020). Polarization and partisanship: Key drivers of distrust in media old and new? *European Journal of Communication*, 35(5):484–501.

[174] Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2018). Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*, pages 2071–2080.

[175] Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., and de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. In *Proceedings of the Workshop on Data Science for Social Good, SoGood 2017*, pages 1–15.

[176] Tandoc Jr, E. C., Lim, D., and Ling, R. (2020). Diffusion of disinformation: How social media users respond to fake news and why. *Journalism*, 21(3):381–398.

[177] Taylor, R. S. (1962). The process of asking questions. *American Documentation*, 13(4):391–396.

[178] Törnberg, P. (2018). Echo chambers and viral misinformation: Modeling fake news as complex contagion. *PloS One*, 13(9):e0203958.

[179] Trokhymovych, M. and Saez-Trumper, D. (2021). WikiCheck: An end-to-end open source automatic fact-checking API based on Wikipedia. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 4155–4164.

[180] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *Proceedings of the ACM International Conference on Machine Learning*, pages 2071–2080.

[181] Trummer, I. (2021). WebChecker: Towards an infrastructure for efficient misinformation detection at web scale. *Data Engineering*, 44(3):66–77.

[182] Tu, M., Wang, G., Huang, J., Tang, Y., He, X., and Zhou, B. (2019). Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713.

[183] Vasileva, S., Atanasova, P., Màrquez, L., Barrón-Cedeño, A., and Nakov, P. (2019). It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. In *Proceedings of the ACL International Conference Recent Advances in Natural Language Processing*, pages 1229–1239.

[184] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 5998–6008.

[185] Visentin, M., Pizzi, G., and Pichierri, M. (2019). Fake news, real problems for brands: The impact of content truthfulness and source credibility on consumers' behavioral intentions toward the advertised brands. *Journal of Interactive Marketing*, 45:99–112.

[186] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

[187] Vrandečić, D. and Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

[188] Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 422–426.

[189] Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1112–1119.

[190] Witten, I. H. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30.

[191] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

[192] Wood, I., Johnson, M., and Wan, S. (2021). Integrating lexical information into entity neighbourhood representations for relation prediction. In *Proceedings of the ACL Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3429–3436.

[193] Workman, M. (2018). An empirical study of social media exchanges about a controversial topic: Confirmation bias and participant characteristics. *The Journal of Social Media in Society*, 7(1):381–400.

[194] Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., and Matsumoto, Y. (2020). Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30.

[195] Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2015). Embedding entities and relations forlearning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*.

[196] Yang, K.-C., Niven, T., and Kao, H.-Y. (2019a). Fake news detection as natural language inference. In *Proceedings of the WSDM 2019 Cup Fake News Classification Challenge in the International Conference on Web Search and Data Mining*.

[197] Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., and Lin, J. (2019b). End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77.

[198] Yilmaz, Z. A., Wang, S., Yang, W., Zhang, H., and Lin, J. (2019). Applying BERT to document retrieval with Birch. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 19–24.

[199] Yoo, J. (2007). Ideological homophily and echo chamber effect in internet and social media. *Student International Journal of Research*, 4(1):1–7.

[200] Zhang, H., Fan, Z., Zheng, J.-h., and Liu, Q. (2012). An improving deception detection method in computer-mediated communication. *Journal of Networks*, 7(11):1811–1816.

[201] Zhang, J., Dong, B., and Philip, S. Y. (2020). FAKEDETECTOR: Effective fake news detection with deep diffusive neural network. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 1826–1829.

[202] Zhang, S., Tay, Y., Yao, L., and Liu, Q. (2019a). Quaternion knowledge graph embeddings. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 2735–2745.

[203] Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019b). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the ACL Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.

[204] Zhao, H., Fu, S., and Chen, X. (2020). Promoting users' intention to share online health articles on social media: The role of confirmation bias. *Information Processing & Management*, 57(6):102354.

[205] Zheng, D., Song, X., Ma, C., Tan, Z., Ye, Z., Dong, J., Xiong, H., Zhang, Z., and Karypis, G. (2020). DGL-KE: Training knowledge graph embeddings at scale. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*, pages 739–748.

[206] Zhou, X. and Zafarani, R. (2019). Network-based fake news detection: A pattern-driven approach. *ACM Knowledge Discovery and Data Mining: Explorations Newsletter*, 21(2):48–60.

[207] Zhou, X. and Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40.

[208] Zhu, G. and Iglesias, C. A. (2015). Sematch: Semantic entity search from knowledge graph. In *Joint Proceedings of the International Workshop on Summarizing and Presenting Entities and Ontologies and the International Workshop on Human Semantic Web Interfaces in the Extended Semantic Web Conference*, pages 1–12.

[209] Zhu, G. and Iglesias, C. A. (2016). Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):72–85.

[210] Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., and Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS One*, 11(3):e0150989.