AN EXPLAINABLE KNOWLEDGE GRAPH BASED MACHINE LEARNING MODEL FOR FACT CHECKING

ARGHYA KUNDU

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

DECEMBER 2023

Abstract

'Fake news' can be broadly categorized into disinformation (intentionally deceiving) and misinformation (unintentionally misleading). For the sake of brevity, we will use misinformation to indicate both types of 'fake news' in this thesis. Misinformation is a growing threat to the economy, social stability, public health, democracy, and national security. One of the most effective methods to combat misinformation is fact checking. Fact checking is the process of verifying the factual accuracy of a statement or claim. Fact checkers employ rigorous methodologies to scrutinize claims, verify sources, and expose falsehoods. However, the huge volume of content circulating online makes it challenging for humans to identify misinformation manually. Automated tools can analyze large datasets to detect patterns in misinformation content, scaling up fact checking efforts.

In this thesis, we propose fact checking methods using natural language processing (NLP) and misinformation propagation patterns. Given a claim, we propose a NLP model that classifies the claim as true or false, and provide explanations for the model's classification. We also propose a classifier that uses propagation patterns of tweets on Twitter to predict whether a tweet is likely to contain misinformation or not, enhancing the classification performance of the NLP model. Specifically, the contributions of this thesis are as follows:

1. We propose a knowledge graph-based fact checking model that uses two separate knowledge graphs (KG), one containing true claims and the other, false claims (misinformation). The model uses knowledge graph embeddings which are based on convolutional neural networks. The deep learning model is trained on the above two knowledge graphs to learn distinguishing patterns between true and false claims. Additionally, we employ explainable artificial intelligence (XAI) techniques to provide explanations for the model's classification, reducing cost of errors and increasing transparency and user trust in the system.

2. We propose a propagation-based classifier to complement the above KG-based fact checking model for misinformation detection on Twitter. The propagation-based misinformation detection model uses the following information for the task: (a) temporal and spatial properties of diffusion cascades of tweets, and (b) "infectiousness" properties of misinformation, which are learned from the spread of COVID-19 and epidemiological models of infectious diseases. Experimental results show that, by combining the propagation-based classifier with the KG-based fact checking model, the prediction of

true/false claims on Twitter achieves higher accuracy compared with the case of using only the KG-based model.

3. We propose a 'translator' program that converts text with slang and non-standard words (SNSW) into standard English for fact checking on Reddit. The 'translated' content is then input into the above KG-based fact checking model, increasing the model's classification accuracy compared with the case where input contains SNSW. 'Translated' text in standard English can also be used by other downstream NLP tasks such as sentiment analysis. Experimental results show that 'translated' text significantly improves the prediction performance of these NLP tasks.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. U.T Nguyen, for all her support, patience and guidance throughout my Master's degree. I offer my gratitude to all committee members for their time and valuable comments that contributed to the improvement of my thesis. I am grateful to all of those I have had the pleasure to work with. Last but not least, thank you to my family members for their endless support and encouragement. To each individual who played a part in this journey, your impact is immeasurable, and I am here because of you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we provide background information and the motivations for our research, define the problems, and discuss the contributions of the thesis.

1.1 Background Information

'Fake news' is a pervasive and serious problem with far-reaching consequences. In general, 'fake news' can be categorized into two major types: misinformation and disinformation [1]. Misinformation is false content shared by a person who does not realize it is false or misleading. Disinformation, on the other hand, is intentionally spread to mislead or deceive. For the sake of brevity, in this thesis we will use misinformation to denote both misinformation and disinformation.

Misinformation poses a threat to democratic processes by manipulating public opinions and influencing elections [2]. When false claims are deliberately spread, it erodes trust in institutions, leading to social divisions and polarization. Furthermore, the negative impact of misinformation extends to public health and safety. Misinformation about health conditions and treatments can result in harmful behavior and a decline in vaccination rates [3]. This issue has been particularly evident during the COVID-19 pandemic, where misinformation has hindered public health efforts and caused confusion among the general public. For example, a rumor caused the death of 800 people after the consummation of alcohol-based cleaning products as a cure for COVID-19 [4].

Thus, misinformation identification is very important. Misinformation identification, specifically through fact checking, plays a crucial role in mitigating the spread of false content, restoring trust in institutions, safeguarding public health, and promoting informed decision making among the general public. Fact checkers help combat the spread of misinformation and promote informed decision making among the public [5]. Fact checking organizations and independent researchers employ rigorous methodologies to scrutinize claims, verify sources, and expose falsehoods, thereby contributing to the mitigation of the impact of misinformation.

Due to the time consuming and difficult nature of manual misinformation identification, the need for automated tools has become increasingly apparent. With the vast volume of

content circulating online, it is challenging for humans to identify misinformation manually. As a result, automated tools leveraging advancements in natural language processing and machine learning have emerged as valuable resources in the fight against misinformation [6]. These tools can analyze large datasets to detect patterns of true/false claims, and identify misinformation at a much faster pace, assisting humans in the process of fact checking. These tools offer the potential to scale up fact checking efforts and provide timely responses to the rapid spread of misinformation, thereby effectively combating the spread of misinformation.

1.2 Motivations

Existing literature on misinformation identification can be broadly categorized into four approaches [7]: knowledge-based, style-based, propagation-based and credibility-based.

Knowledge based approaches aim to use external sources to verify statements made in news content [8], [9], [10]. A knowledge-based approach in misinformation identification is fact checking. Fact checking is the process of verifying the factual accuracy of a statement or claim. The goal of fact-checking is to assign a truth value to a claim in a particular context [6], [11]. By using natural language processing techniques, inconsistencies, language patterns, and contextual clues can be identified. Fact checking approaches offer a direct analysis of the content and have proven effective in misinformation identification [12], [11], [13].

Style-based approaches in misinformation detection focus on analyzing the specific writing style employed in fake news articles [14], [15], [16]. This includes examining the language, symbols, and overall structure used in the content. These methods operate under the assumption that the distribution of words and expressions in fake news significantly differs from that of truthful news [7]. Another approach is stance detection, which is focused on analyzing the stance or perspective expressed in news articles [17], [18]. For example, identifying whether a particular news headline 'agrees' with, 'disagrees' with, 'discusses,' or is 'unrelated' to a particular news article allows journalists and others to find and investigate possible instances of misinformation more easily [19]. However, the weakness of stance detection lies in the subjectivity of determining the true stance and potential misinterpretation of the author's intent.

Propagation-based methods study the spread of misinformation on social media platforms, for example, the dynamics of the spread of a tweet on Twitter [20], [21]. By learning the propagation cascades of a post and dynamics of information dissemination these methods infer the veracity of the tweet [22]. However, these methods may overlook instances of targeted and covert dissemination that do not lead to widespread propagation.

Credibility-based methods evaluate the credibility of content by considering the reputation or authority of the source [23], [24], [25]. These models use classifiers or network analysis techniques to assess source credibility. However, relying solely on source credibility may not be sufficient to identify cases where previously credible sources disseminate false claims. While the reputation of a source can serve as an initial indicator of credibility, it is essential to recognize that even reputable sources can occasionally publish misleading or inaccurate content.

Relying solely on source credibility and propagation information may not be sufficient, and these techniques should be used in conjunction with other methods for accurate misinformation identification [26], [27], [23]. Among the various mentioned methods, the fact checking approach is the most effective for misinformation identification due to its ability to directly analyze the characteristics and language patterns of the content itself [28], [29]. Additionally, while these misinformation identification methods and tools have shown promise, challenges remain in developing robust models that can effectively analyze the nuanced language, subtle biases, and contextual factors present in misinformation.

1.2.1 Fact Checking of Claims

In this thesis, we define fact checking as follows: Given a claim, we classify it as true or false. For instance, given a claim, *"Scientists have discovered that COVID-19 can be cured by eating a combination of raw onions, lemon juice, and crushed aspirin."*, our fact checking model classifies the claim as false.

Existing methods of machine learning based fact checking employ a range of techniques, including language analysis [30], [31], evidence retrieval techniques [32], [33] and knowledge graphs to assess the veracity of claims [8], [34]. These approaches use different sources of evidence for fact checking.

Language analysis techniques for fact checking operate under the assumption that the language used in a claim can provide indications of its veracity. Studies have shown that false claims often rely on differences in phrasing rather than outright fabrications, making language analysis a useful tool [16]. For example, subjective words can be used to dramatize or sensationalize a story. This information has been used to develop several models that take a claim and term frequency-inverse document frequency (TF-IDF) data as input and predict the veracity of the claim [34]. Furthermore, the creators of fake news take into account various stylistic tricks, including evoking emotional responses in recipients. This has led to the use of sentiment analysis to assess the subjective tone and predict the veracity of textual claims [35]. Researchers have also explored few-shot learning with pre-trained large language models to capture language nuances and patterns for fact checking purposes [31].

However, language analysis techniques have their limitations. They often focus on analyzing individual sentences or phrases without considering the broader context, leading to potential misinterpretation and incomplete understanding of intended meaning, resulting in inaccuracies in fact checking outcomes [34], [36]. Additionally, language analysis struggles to capture nuances such as sarcasm, irony, or subtle expressions, which can be used to convey misinformation or misleading statements, posing challenges for fact checking systems. While sentiment analysis can provide insights into subjective aspects, it does not inherently verify the factual accuracy of a claim. Fact checking requires additional evidence and contextual information beyond sentiment analysis. Moreover, adversaries can intentionally manipulate language or exploit vulnerabilities in language analysis models, compromising the reliability and effectiveness of these techniques for fact checking purposes [36].

Evidence retrieval (ER) techniques retrieve relevant documents or articles from the web in

real time that contain evidence related to a claim and use them as ground truth to determine the veracity of the claim [34], [37]. ER methods can involve techniques like question generation, keyword matching, document similarity scoring, and document ranking [38] and use search engines like Google, Bing or curated list of websites and resources for retrieving the evidence [34].

However, ER techniques have certain limitations. One limitation is the potential for incomplete or unreliable evidence sources being retrieved, which can compromise the accuracy of fact checking results. Furthermore, ER techniques may face challenges when dealing with complex claims that require multiple pieces of evidence or reasoning. Therefore, it is crucial to supplement ER techniques with evidence selection and verification methods to ensure the production of accurate fact checking outcomes [10], [34]. Additionally, in cases where misinformation campaigns intentionally manipulate claims, evidence retrieval methods may struggle to identify reliable and trustworthy sources. Misinformation can be strategically disseminated across various platforms, making it challenging to gather relevant and accurate evidence.

Among these methods, one approach that has gained significant attention is the use of knowledge graphs [8], [39]. Knowledge graphs are structured representations of facts that capture relationships between entities and their attributes. They organize knowledge in a graph-like structure, where entities are represented as nodes and relationships as edges, enabling efficient retrieval and inference of information. In the context of knowledge graphs, a triplet is a fundamental unit of information, consisting of three components: subject, predicate, and object. For instance, consider the triplet "Albert Einstein - was born in - 1879." Here, "Albert Einstein" serves as the subject, "was born in" represents the predicate, and "1879" is the object. Triplets form the building blocks of knowledge graphs, facilitating the representation and organization of complex information.

The knowledge graph approach for fact checking involves the construction and use of a graph-based representation that organizes factual knowledge and its relationships. By using the structured data, machine learning algorithms can navigate through the graph to extract relevant facts and assess the veracity of claims. They use graph embedding or graph pattern mining methods to learn the features or rules that can distinguish between true and false claims [40]. One advantage of the knowledge graph approach is its ability to capture complex relationships and dependencies between entities [40]. By representing information in a structured manner, knowledge graphs enable the identification of implicit connections that may not be evident from individual sources or isolated statements. Furthermore, knowledge graphs can be continuously updated and enriched with new information, ensuring the model remains adaptable to evolving misinformation and disinformation.

Previous research has demonstrated the potential of knowledge graphs in improving fact checking. For instance, Shiralkar et al. [41] used a method that flows information from the subject to the object through different paths in the graph and compares the amount of information with a threshold to decide the truthfulness. Shi et al. [39] proposed a method that finds different paths from the subject to the object in the graph and ranks them based on how well they can separate true and false claims. The paper then combines the ranked

paths into a feature vector and feeds it to a binary classifier to predict the veracity of the content.

Nevertheless, existing models use knowledge graphs build from only true statements (truthful information). However, by using additional knowledge graphs built from false content, a wider range of information and relationships can be captured, resulting in a more complete understanding of the subject matter, and making it easier to detect misinformation. Limited research has been conducted in this direction.

Pan et al. [42] used two knowledge graphs build from fake news and true news and trained two TransE models on the knowledge graphs for fake news detection. They used the max bias of triplets for detecting the veracity of news articles. However, their approach of using news articles to construct the knowledge graphs has limitations. They consider the triplets from the entire 'fake news' articles to construct the 'fake news' knowledge graph. However, such an article also includes true claims, resulting in the inclusion of the true claims in the fake news knowledge graph. This can create confusion and inaccuracies in the representation of knowledge. Additionally, their use of a translational embedding model (TEM) has several limitations that make it less suitable for fact checking. Specifically, TEM struggles with modeling complex relationships beyond simple translations. It is unable to handle one-to-many or many-to-one relations, as well as symmetric relations [43, 9]. Fact checking on the other hand, requires addressing inconsistencies, understanding complex relationships, and performing inference. For these purposes, more advanced deep learning-based embedding models that incorporate logical reasoning and semantic constraints are more appropriate for fact checking.

In this thesis, we develop a fact checking model that uses knowledge graphs to verify the veracity of claims. To achieve this, we propose a model consisting of two knowledge graphs: one created using true claims and the other created using false claims. Two separate deep learning models are trained on these two knowledge graphs using convolutional neural networks (CNN)-based knowledge graph embeddings, enabling the models to learn distinguishing patterns and characteristics of true and false claims.

Additionally, one of the key challenges in fact checking is the lack of explanations and rationales accompanying the model's predictions. Many existing machine learning models provide labelled output without offering insights into the rationales and explanations behind the decision of a model. This lack of transparency often leads to reduced user trust, as users may struggle to understand the rationale behind the model's predictions [44]. With the advance of AI techniques, current AI systems address sophisticated real-world problems but inevitably become black boxes, yielding opaque results, which are difficult to understand [45]. The undesired black-box model problem has developed into a growing research field, explainable artificial intelligence (XAI), in the last few years. XAI aims to provide models with the capability to not only produce predictions but also offer explanations for their predictions [46]. In our work, we provide explanations for the predictions generated by the models, to reduce cost of errors and increase user trust in the system.

Given a claim, *"Earth has not warmed for the last 17 years."*, our model classifies the claim as false. Additionally, it points out the false (or true) triplet(s) in a claim in a colour

coded manner, and provides the confidence scores for each triplet. Figure 1.1 shows a sample output of our base model given a claim along with the triplet and confidence scores.

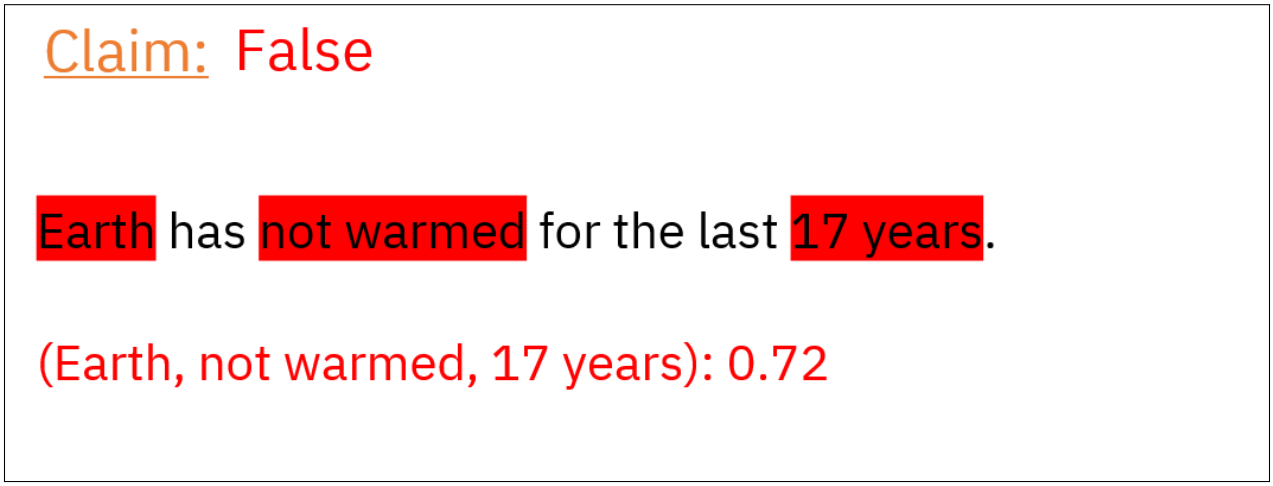


Figure 1.1: Sample output of our model given a claim

1.2.2 Propagation of Claims in Social Media

The rise of social media platforms like Twitter and Reddit has significantly impacted the way people consume news. Nowadays, a large portion of the population relies on social media platforms to stay updated with current events. According to a survey conducted by the Pew Research Center (2021) on news consumption habits, approximately 53% of respondents indicated that they primarily obtain their news from social media sources, representing a significant increase from previous years [47].

While social media provides a convenient and accessible platform for news dissemination, it also presents a significant challenge in terms of credibility and reliability of news shared. One of the major concerns associated with social media is the proliferation of misinformation and disinformation. Misinformation spreads rapidly through online social networks, often leading to misunderstandings, confusion, and sometimes even harmful consequences. Considering the prevalence of misinformation on online social networks, the need for effective fact checking mechanisms on these platforms has become increasingly critical. Such mechanisms are essential for ensuring the accuracy and authenticity of the claims circulating on these platforms.

Online social networks have additional information, such as comments, user emotions, and propagation patterns, that can be used in conjunction with content-based fact checking methods to increase the accuracy of misinformation identification on such platforms. Analyzing user comments provides insights into community sentiment and skepticism, helping identify potential falsehoods. User emotions serve as indicators of emotional manipulation tactics used in misinformation. While comments and user emotions offer insightful cues, the analysis

of propagation patterns emerges as a particularly influential factor. Studying propagation patterns, allows for the identification of suspicious or abnormal dissemination patterns [20], [21].

In a study funded by Twitter, Vosoughi et al. [48] found that diffusion cascades of misinformation are different from those of truthful news. The authors concluded that misinformation propagates significantly farther, faster, deeper, and broader than truthful news in many categories. Recently Juul et al. [49] repeated the comparison of the dissemination of fake and true news conducted by Vosoughi et al. on the same dataset. Although Juul et al. confirmed the basic conclusion of Vosoughi's study rather than contradicting it, they concluded that the four qualities — further, faster, deeper, and broader - can be attributed to the "infectiousness" of misinformation. The study used two models, namely, the SIR (Susceptible-Infectious-Recovered) and IC (Independent Cascade) models and found that the differences in diffusion can be explained by differences in "infectiousness" alone. They concluded that fake news is simply more "infectious" than truthful news.

While both studies provide empirical evidence that fake news can be differentiated from truthful news using propagation cascades, it remains unclear how this difference emerges and how it can be properly used to create verifiable automated detection mechanisms. Additionally, researchers have attempted to construct automated fact checkers, relying primarily on the work by Vosoughi et al. without considering the factor of "infectiousness" [50], [20].

In this thesis, we exploit propagation patterns of tweets on Twitter to enhance the classification accuracy of the knowledge graph-based model described above for misinformation identification on Twitter. Given a claim (post) circulating on Twitter, we propose a propagation-based model to classify the veracity of the claim. The proposed model is an ensemble classifier which uses the diffusion pattern of a tweet in Twitter. The ensemble classifier uses the following information for the task: (a) temporal and spatial properties of diffusion cascades of tweets, and (b) the "infectiousness" properties of misinformation which are learned from the spread of COVID-19 and epidemiological models of infectious diseases.

We then combine this propagation-based model with the the knowledge graph-based model described previously for classification of Twitter posts. Experimental results demonstrate that the combined model significantly increases the accuracy of Twitter post classification compared to using each model independently.

Furthermore, we use the knowledge graph-based classifier to facilitate fact checking on Reddit. Reddit and Twitter, though both influential platforms in the realm of digital communication, present different challenges and dynamics when it comes to the dissemination of information. While Reddit boasts a blog-like, topic-centric structure, Twitter leans towards a user-centric model. These core differences play a crucial role on how information flows and the inherent challenges tied to fact checking within each environment.

Conversational language in platforms like Reddit, which resemble blogs, frequently contains jargon, slangs specific to Reddit, and colloquial terms [51, 52]. One of the challenges in NLP is the ability to understand and process jargon, slangs, and colloquial terms, also known as out-of-vocabulary(OOV) words. Conversational language used in blogs often contains such slang and non-standard words (SNSW). SNSW are not typically present in curated

datasets used to train NLP models. As a consequence, it is difficult for traditional models to accurately interpret and analyze text data with SNSW [53, 54]. Recognizing the challenges posed by SNSW, we introduce a transformer-based translator that converts text with SNSW into standard English. The 'translated' content is then input into the above KG-based fact checking model, increasing the model's classification accuracy compared with the case where input contains SNSW. 'Translated' text in standard English can also be used by other downstream NLP tasks such as sentiment analysis.

1.2.3 Explainability of Machine Learning Models

The need for transparency in modern machine learning methods has become evident following events such as the Cambridge Analytica scandal and the disruptions of the 2016 US elections [55]. This has led to the launch of initiatives like DARPA's eXplainable AI program [56] and the European Union Ethical guidelines for Trustworthy AI [57], which aim to promote the design of ethical systems that humans can understand, manage, and trust. Explainable Artificial Intelligence (XAI) allows models to provide interpretable results, which can be explained through various techniques such as highlighting specific parts of the input, contrasting with similar instances, or using natural language. Interpretability is crucial because many machine learning models are being deployed without users understanding the logic behind the model's predictions [56], [58], thus creating a lack of trust between the users and the model. Moreover, XAI helps designers gain insights into the reasoning process and improve the model.

In the domain of misinformation identification, when a model assists a user in determining the trustworthiness of an article, the user's trust in the model significantly impacts their judgment of the content. Regardless of the model's effectiveness in predicting the veracity, the user ultimately decides whether to believe the claims presented in the article. When the user lacks an understanding of the model or perceives it as a black box, their trust is generally low [46], [58]. However, striking a balance between explainability and performance is a major challenge. As machine learning models become increasingly complex and successful in solving learning problems, their inherent lack of transparency poses a significant hurdle in making them explainable for XAI [58]. This tradeoff between complexity and explainability highlights the need for developing techniques and methodologies that can provide insights into the decision making process of complex machine learning models without compromising their performance.

In this work, we incorporate explainability while maintaining high accuracy in our models. We present two explainable models: a knowledge graph-based fact checking model and a propagation-based tweet verification model, both of which provide rationales behind their predictions to the users. In the knowledge graph-based checking model, we utilize a color-coded scheme to provide explanations for true and false triplets in a claim. This scheme helps users understand the basis for the model's predictions by visually highlighting the supporting or contradicting evidence from the underlying knowledge graph. On the other hand, the propagation-based tweet verification model generates explanations by using spatio-temporal

and epidemiological features. These features allow us to provide insights into the factors that contribute to the model's predictions.

By incorporating these explanatory mechanisms, our work will not only helps users make informed judgments about the veracity of claims, but also enable them to trust and understand the reasoning behind the model's predictions.

1.3 Contributions

The main contributions of the thesis are as follows:

1. We propose a knowledge graph-based fact checking model that uses two separate knowledge graphs (KG), one containing true claims and the other, false claims (misinformation). The proposed model is trained by using ConvE, a CNN-based knowledge graph embedding method. Experimental results show that our model achieves significantly higher classification performance compared to six baseline misinformation detection models. Furthermore, the inclusion of false claims knowledge graphs increases the accuracy of fact checking, compared with the case of using only true claim knowledge graphs. We employ XAI techniques to provide explanations for the model's classification. The ConvE model is used to generate confidence scores for individual triplets. The triplets are highlighted with color red for false claims, and green for true claims. The XAI output reduces cost of errors, and enables end-users to understand the rationales behind the model's predictions.

2. We propose a propagation-based classifier to complement the above KG-based fact checking model for misinformation detection on Twitter. The propagation-based misinformation detection model uses the following information for the task: (a) temporal and spatial properties of diffusion cascades of tweets, and (b) "infectiousness" properties of misinformation, which are learned from the spread of COVID-19 and epidemiological models of infectious diseases. Experimental results show that, by combining the propagation-based classifier with the KG-based fact checking model, the prediction of true/false claims on Twitter achieves higher accuracy compared with the case of using only the KG-based model. Furthermore, explanations based on the diffusion pattern of a tweet are provided to illustrate the rationales behind the propagation-based model's predictions.

3. We propose a transformer-based 'translator' program that converts text with slang and non-standard words (SNSW) into standard English for fact checking on Reddit. The 'translated' content is then input into the above KG-based fact checking model, increasing the model's classification accuracy compared with the case where input contains SNSW. 'Translated' text in standard English can also be used by other downstream NLP tasks such as sentiment analysis. Experimental results show that 'translated' text significantly improves the prediction performance of other NLP tasks.

Overall, these contributions advance the field of automated fact checking and misinformation detection, and enhance the transparency and user trust via XAI.

1.4 Thesis Organization

The remainder of the thesis is organized as follows:

Chapter 2: Literature Review
In this chapter, we provide a comprehensive review of existing approaches for fact checking and misinformation identification. We explore various techniques, methodologies, and datasets related to these research areas. This review serves as a foundation for understanding the current research state of the field and identifying the research gaps that our thesis aims to address.

Chapter 3: Fact Checking Model Using Knowledge Graphs
In this chapter, we introduce the proposed knowledge graph-based fact checking model. We describe the methodology, architecture, and training process of the NLP model that uses two knowledge graphs created using true and false claims. We also highlight the generation of explanations by the NLP model, which aims to enhance the interpretability and transparency of the model's predictions

Chapter 4: Fact Checking on Twitter Using Diffusion Dynamics and Knowledge Graphs
We propose a propagation-based misinformation detection model specifically designed for tweets on Twitter, which leverages diffusion statistics to classify posts on Twitter as true or false. We discuss how the model generates explanations based on the diffusion pattern of a tweet to justify the predictions. We combine the propagation-based model with the above knowledge graph-based fact checking model to fact check claims on Twitter.

Chapter 5: Fact Checking on Reddit Using Knowledge Graphs
Recognizing the challenges posed by SNSW, we develop a 'translator' that converts text with SNSW into standard English. The output of the 'translator' is used as the input to the above knowledge graph-based fact checking model to fact check claims on Reddit. We also demonstrate the application of the 'translator' to other downstream NLP tasks such as sentiment analysis.

Chapter 6: Conclusion and Future Work
In the final chapter, we summarize the key contributions of our thesis and present future research directions in the topic of automated fact checking.

Chapter 2

Literature Review

In this chapter, we provide a comprehensive review of existing work related to fact checking, knowledge graphs, and misinformation identification. By examining previous research, methodologies, and approaches, we aim to establish a solid foundation for our study and identify the gaps in existing research.

2.1 Misinformation Identification

Existing literature that studies misinformation identification can be categorized into four groups [7]: (i) knowledge-based, (ii) style-based, (iii) propagation-based and (iv) credibility-based. In this section, we will examine each category. Following this, the next section will concentrate on fact-checking methods, which are an effective approach for misinformation identification.

2.1.1 Knowledge-based Approach

A knowledge-based approach in misinformation identification is fact checking. Knowledge based approaches aim to use external sources to fact check proposed claims in news content [8], [9], [10]. The goal of fact-checking is to assign a truth value to a claim in a particular context [6]. Fact checking has attracted increasing attention, and many efforts have been made to develop a feasible automated fact checking system. Existing knowledge based fact checking approaches can be categorized as expert-oriented, crowdsourcing-oriented, and computational-oriented [6].

Expert-oriented fact-checking heavily relies on human domain experts to investigate relevant data and documents to construct the verdicts of claim veracity. Expert-based manual fact checking provides accurate results since fact checkers follow rules when verifying news contents. For example, all fact checkers who are a member of the International fact checking network (IFCN) follows the code of principles for nonpartisan and transparent fact checking [40].

Crowdsourcing-oriented fact checking exploits the "wisdom of crowd" to enable normal people to annotate news content; these annotations are then aggregated to produce an overall assessment of the news veracity. However, this approach is less accurate than the expert-based manual fact checking approach due to conflict annotations. Computational-oriented fact checking aims to provide a machine learning based scalable system to classify true and false claims. An overview of these studies will be provided in Section 2.2.1.

2.1.2 Style-based Approach

Style-based approaches in misinformation detection focus on analyzing the specific writing style employed in fake news articles [14], [16]. This includes examining the language, symbols, and overall structure used in the content. These methods operate under the assumption that the distribution of words and expressions in fake news significantly differs from that of real news [7]. Granik [14] discovered similarities between fake news and spam emails, such as the presence of numerous grammatical errors, attempts to manipulate reader opinions, and the use of a limited set of words. Based on these similarities, they implemented a simple approach for fake news detection using a naive bayes classifier.

Earlier research papers on fake news identification adopted the term frequency-inverse document frequency (TF-IDF) technique to encode the headline and body of news articles separately, a process known as stance detection [17]. This approach involves developing a probabilistic model that captures the language patterns found in fake news articles by counting the frequency of specific words across a range of documents and normalizing it by the total number of documents in which the word appears.

News articles cover a wide range of topics, such as politics, health care, economics, the financial market, and climate change and the identification of misinformation in various fields is different [59]. Therefore, analysis of fake news should be cross-domain, cross-language, and cross-topic.

2.1.3 Propagation-based Approach

Another approach has emerged recently, focusing rather on the propagation of misinformation on social media as a measure of its veracity [20], [21]. Some studies focused on capturing the temporal traits the propagation of a rumour. Kwon [22] introduced a time-series-fitting model based on the temporal properties of a single feature – tweet volume. Ma [60] extended the model using dynamic time series to capture the variation of a set of social context features over time. Friggeri [61] characterized the structure of misinformation cascades on Facebook by analyzing comments. Nevertheless, some kernel-based methods were exploited to model the propagation structure. Wu [62] put forward the idea of a hybrid SVM classifier combined with an RBF kernel and a random walk based graph kernel to encapsulate propagation patterns for detecting rumors on Sina Weibo. Similarly, Ma [63] using a tree kernel, utilized the count of similar substructures to capture the similarity of diffusion cascades and identify rumours on Twitter. However, in addition to the propagation pattern Wu [62] considered

the message topic, sentiments of the responses and the profile information of the users and Ma [63] used the creator and the text content of the post. Furthermore, these studies do not effectively the relevance of the spread of a rumour to that of an infectious disease.

Based on a landmark study funded by Twitter, Vosoughi [48] found that the diffusion cascades of fake news are different from that of real news. The authors concluded that fake news propagates significantly farther, faster, deeper, and broader than true news in many categories of information. However, recently Juul [49] repeated the comparison of the spread of fake and true news conducted by Vosoughi on the same dataset. Even though the authors confirmed the basic conclusion of the Vosoughi's study rather than contradicting it, they figured the four qualities—further, faster, deeper, broader can be attributed to the "infectiousness" of the claims. Juul used two epidemiological models, namely SIR and IC models for study and found that the differences in diffusion can be explained by differences in "infectiousness" alone. They concluded that fake news is simply more "infectious" than the truth.

Researchers have tried and failed to build automated fact checkers, based on the work from Vosoughi alone and not taking "infectiousness" into account [50], [20]. In our work along with the temporal and spatial properties of diffusion cascades, we particularly consider the "infectiousness" property of claims specially by using insights learned from the spread of COVID-19 and general epidemiological models of infectious diseases.

2.1.4 Credibility-based Approach

The credibility-based approach is to detect misinformation by assessing the credibility of the publisher or author of a given statement or an article [23], [24], [25]. Agarwal [24] conducted an analysis of various classifiers for fake news detection, specifically focusing on assessing the credibility of the sources. Their findings revealed that the support vector machine (SVM) and logistic regression classifier performed the best. In a different approach, Yuan [64] introduced a novel network called the structure-aware multi-head attention network (SMAN), which incorporates the news body, publishing, and reposting relations of publishers and users to evaluate credibility. Baruah [25] employed the BERT model to classify whether the author of a Twitter feed is a fake news spreader. Similarly, Yang [65] proposed an unsupervised framework utilizing a probabilistic graph model and the Gibbs sampling approach to estimate user credibility based on user engagement behavior.

However, it is important to note that relying solely on source credibility is not sufficient for accurate misinformation detection. Malicious websites can sometimes contain true news, while reputed news websites may occasionally present false claims. Therefore, these techniques should be used in conjunction with other misinformation detection methods to enhance the overall effectiveness of the detection process.

2.2 Fact Checking

Fact checking is a critical task in the field of claim verification and credibility assessment. With the proliferation of misinformation and disinformation in today's digital age, it has become increasingly important to have robust methods and systems in place to verify the accuracy of claims and check the validity of facts. Fact checking involves examining and verifying the factual accuracy of various pieces of claims contained in statements, claims [6], [11].

2.2.1 General Approaches

Traditional fact checking processes typically involve manual verification by expert fact checkers. These experts examine claims, gather evidence, and assess the accuracy of the claims. However, manual fact checking is a time consuming and resource intensive process, limiting its scalability in handling the vast amount of claims circulating in today's digital age [66].

To address these limitations, researchers have developed automated fact checking systems that employ natural language processing (NLP) techniques and machine learning algorithms [40], [67]. These systems aim to automatically assess the truthfulness of claims by analyzing textual data and comparing it against reliable sources of information, such as reputable news articles or official databases [16], [30]. Crowdsourcing has also emerged as a popular approach in fact checking, where a large group of individuals collaboratively assesses claims and provides their judgments, like Fiskkit.com. These methods uses the wisdom of the crowd to obtain diverse perspectives and opinions, increasing the robustness of the fact checking process [68].

Furthermore, fact checking organizations have established dedicated platforms (e.g., PolitiFact, Snopes, Reuters, and Poynter) to document and disseminate verified claims. These platforms serve as valuable resources for fact checkers, researchers, and the public to access reliable and credible information [2].

Additionally, the advancements in computational techniques, such as machine learning and NLP, have enabled the development of fact checking models that can analyze claims at scale. These models leverage various features, such as claim content, context, source reliability, and historical data, to determine the veracity of the claims [11], [13].

Augenstein [29] introduced a fact checking model that used neural networks to jointly model the claim, evidence, and metadata features. The model learned to encode the input information and make veracity predictions based on the extracted features, improving the accuracy of fact checking.

Nakashole and Mitchell [69] combined linguistic features on the subjectivity of the language used with the co-occurrence of a triplet with other triplets from the same topic, a form of collective classification. Ferreira and Vlachos [18] modeled fact checking as a form of recognizing textual entailment (RTE), predicting whether a premise, typically part of a news article, is related to a given claim. The same type of model was used by most of the 50

participating teams in the Fake News Challenge [70], including most of the top entries such as Riedel [17] and Hanselowski [71].

Despite these advancements, challenges persist in fact checking. The rapid spread of claims through social media platforms poses challenges in timely fact checking, as false claims can quickly gain traction before they are debunked [15], [72]. Additionally, one open issue is the generation of explanations from the machine learning models being used for fact checking [67].

2.2.2 Evidence-based Fact Checking

Evidence based fact checking is a process of verifying the truthfulness of a claim by comparing it with relevant and reliable sources of information. It is often used to assess the accuracy of statements made by politicians, journalists, celebrities, or other public figures.

A recent study [32] introduced a new dataset for evidence-based fact checking of health related claims called HEALTHVER. The dataset contains real-world claims that were retrieved from snippets returned by a search engine for questions about COVID-19. The dataset also contains relevant scientific papers that were automatically retrieved and re-ranked using a T5 relevance-based model. The relations between each evidence statement and the associated claim were manually annotated as 'Support', 'Refute', or 'Neutral'. The authors developed baseline models based on pre-trained language models and showed that training them on real-world medical claims greatly improves performance compared to models trained on synthetic and open-domain claims. Another large-scale dataset for evidence-based fact checking of factual claims is the Fact Extraction and VERification (FEVER) dataset [11]. The dataset contains synthetic claims that were generated by mutating sentences from Wikipedia pages. The dataset also contains evidence sentences that were extracted from Wikipedia pages using an information retrieval system. The relations between each claim and evidence set were manually annotated as 'Supports', 'Refutes', or 'Not enough info'. The authors presented several baseline models based on neural networks and showed that the task is challenging and requires complex reasoning skills.

Zhang [33] introduced another dataset and a model for verifying the truthfulness of answers in product related community question answering (PQA) platforms. The dataset, called AnswerFact, contains 60,864 answer claims across five product domains, each with a veracity label and associated evidence sentences from product information. The authors proposed a neural model with tailored evidence ranking module to tackle the answer veracity prediction problem. Additionally, evidence data for fact checking can be in multiple modalities such as text, tables, images, audio, or video. For instance, Akhtar [37] proposed a chart-based fact checking model and introduce ChartBERT for fact checking against chart evidence. ChartBERT uses textual, structural, and visual information of charts to determine the veracity of textual claims.

Evidence-based fact checking is an important and difficult task that requires finding and analyzing relevant sources of information. One of the main challenges of evidence-based fact checking is finding reliable and relevant sources of information. There is a lot of claims

available online, but not all of it is trustworthy or accurate. Some sources may be biased, outdated, incomplete, or even deliberately misleading. It can be hard to verify the credibility and quality of the sources, especially when they contradict each other or have different perspectives on the same topic. Another challenge of evidence-based fact-checking is dealing with complex and ambiguous statements. Some statements may not have a clear or definitive answer, but rather depend on the context, interpretation, or opinion of the speaker or the listener.

Knowledge graph-based fact checking can overcome some of these challenges by using a large and structured collection of facts and concepts that are linked by semantic relationships [8], [10]. A knowledge graph can represent a wide range of domains and topics and provide a rich and consistent representation of the world's knowledge. Although evidence-based fact checking is generally considered more accurate and trustworthy, it can be a time consuming and labor-intensive process. However, knowledge graph-based fact checking offers the advantage of requiring less time and computational effort compared to evidence-based approaches [10].

2.2.3 Knowledge Graph-based Fact Checking

Knowledge graphs have emerged as valuable resources for various natural language processing tasks, including fact checking [10]. A knowledge graph represents structured knowledge about entities, their attributes, and relationships between them. In the context of fact checking, knowledge graphs provide a rich source of information that can be used to assess the truthfulness of claims [8], [9].

Knowledge graph-based fact checking systems uses the structured knowledge within knowledge graphs to enhance the accuracy and explainability of the fact checking process. By capturing relationships between entities, knowledge graphs enable a more nuanced understanding of claims and their context. This enables the verification of claims by accessing reliable and relevant information.

For example, Ciampaglia [8], proposed a weighted knowledge stream/segment based method to check the truthfulness of a given claim. They used a TF-IDF strategy to transform the knowledge graph into a weighted graph, then made the predication according to the knowledge stream. Shi [39] proposed a discriminative paths based method using knowledge graphs to predict the truthfulness of a given claim. Their method used a supervised learning approach to rank the paths based on their discriminativeness and then combines them into a feature vector for a binary classifier. Vlachos and Riedel [13] introduce a dataset and task for fact checking, highlighting the importance of using knowledge graphs as a resource for verifying claims. Shiralker [41] proposed a fact checking algorithm called Relational Knowledge Linker, which converts the knowledge network to a smooth–continuous network and examines a claim on the single shortest and semantically connected path in the knowledge graph.

Pan [42] used two knowledge graphs to train separate models on fake news and true news for fake news classification. Among the existing literature, while our approach for fact checking using knowledge graphs (KG) incorporates unique elements that distinguish it from

other methods discussed in the literature, it shares similarities with the study conducted by Pan [42]. Our approach, like theirs, acknowledges the importance of using two knowledge graph embedding models trained on two separate knowledge graphs. However, Pan's approach has several limitations.

1. Their use of a translational embedding model (TEM) has limitations that make it less suitable for fact-checking tasks involving knowledge graphs. TEM struggles with modeling complex relationships beyond simple translations, and it is unable to handle one-to-many or many-to-one relations, as well as symmetric relations [43, 9]. Its assumption of representing relations as translations restricts its expressiveness, making it difficult to capture nuanced or hierarchical relationships accurately [9]. In addition, TransE lacks mechanisms to handle contradictions, or effectively deal with sparse data [73]. For fact-checking, which requires addressing inconsistencies, understanding complex relationships, and performing inference, more advanced deep learning based embedding models like CNN-based models incorporating logical reasoning, semantic constraints are more appropriate. Our study address this limitation by using ConvE, a CNN-based model.

2. For classification the authors used only one feature (the bias) from the trained model. However, our study shows that using additional knowledge graph embedding-specific data can improve accuracy.

3. Their study used news articles to construct the knowledge graphs which presents a challenge. Fake news may contain true claims, resulting in the inclusion of such information in the fake news knowledge graph. This mixture of inaccurate claims in the same knowledge graph introduces noise and potential confusion, making it more challenging to distinguish between true and false claims. However, our study uses claims (one or two sentences) instead of news articles for creating knowledge graphs thus limiting the introduction of noise.

4. The study performed binary classification (True/Fake) of news articles. However, news articles often exhibit a spectrum of veracity, ranging from completely true to completely false, with various degrees of accuracy in between. This is because news articles often contain multiple claims, which can either be true or fake.

5. An important limitation of their model is the lack of inclusion of updated knowledge. Fact checking is an ongoing process, and new information emerge constantly. Our model regularly integrating current data from various fact checking websites.

6. The model employed in their study lacks the generation of explanations, which is an essential component for transparency in fact checking systems. Providing explanations enhances user understanding and trust in the fact checking process. Our models provide explanations along with the classification result.

2.3 Explainability of Fact Checking Models

Approaches to explainable machine learning are generally classified into two categories: post-hoc explainability and intrinsic explainability [46]. Intrinsic explainability is achieved by constructing self-explanatory models which incorporate interpretability directly to their structures. In contrast, the post-hoc interpretability requires creating a second model to provide explanations for an existing model which is considered as a black-box.

Previous works have demonstrated the potential of these approaches in enhancing the explainability of fact checking models [46], [74], [75]. For instance, Khoo [12] used post-level and comment-level attention mechanisms to identify the most important tweets related to a target news article. They also highlighted the most significant words in those tweets, providing users with insights into the relevant content. Whereas, Chen [76] incorporated both latent and handcrafted features of news content and responses, along with user features, to determine the authenticity of news. Through attention mechanisms, they were able to highlight the most important features of news articles, offering explanations for their results.

In studies such as Reis [77], various models are trained, each with a different set of features. The most important features of the best-performing models are then extracted using a method called Shapley additive explanations (SHAP) [78]. This technique allows for the identification of the key features influencing the model's decision. Some approaches combine multiple models to offer different explanations. Mohseni [74] trained four different models, each providing a distinct explanation for classifying news articles. Similarly, Yang [67] developed the XFake model, which comprises three frameworks: MIMIC, ATTN, and PERT. Each framework provides a unique explanation by focusing on different aspects of the news.

While numerous studies underscore that algorithmic interpretability can enhance user trust in algorithms, many misinformation detection models overlook the importance of explainability [46], [74]. Furthermore, research suggests that intrinsic explainable artificial intelligence (XAI) methods yield superior explanations compared to post-hoc XAI techniques [79], even though they may sometimes compromise accuracy. However, much of the aforementioned research tends to treat models as black-boxes, predominantly leaning towards post-hoc explainability. Diverging from this trend, our study integrates intrinsic explainability directly into the models without compromising accuracy. Additionally, our approach introduces a confidence score, amplifying the transparency and interpretability of the results.

Chapter 3

Fact Checking Model Using Knowledge Graphs

Misinformation identification approaches encompass a variety of techniques including language analysis and stance detection, among others. However, among these methods, the fact checking approach stands out as the most effective for identifying misinformation, owing to its direct analysis of the characteristics and content of the information. This chapter discusses the data structures of knowledge graphs and the proposed knowledge graph-based fact checking (KG-FC) model. It presents an evaluation of the proposed model using various metrics and compares it with other state-of-the-art misinformation detection models.

The chapter is structured as follows: Section 3.1 discusses the importance of misinformation identification and the motivation of our work. Section 3.2 describes knowledge graphs, knowledge graph embeddings, the ConvE embedding technique and the training assumptions used in the study. In Section 3.3, we present the methodology used by our model. Section 3.4 discusses the explainability of the model. In Section 3.5 we discuss the dataset used for the study and present the evaluation our model using different metrics, followed by the ablation study and the comparison of our model with different state-of-the-art models. Section 3.6 lists the limitations of the model. Finally, we summarize the chapter in Section 3.7.

3.1 Introduction

Knowledge graphs (KG) are data structures that effectively represent information from diverse domains using triplets (h, t, r), where 'h' denotes the head entity, 't' represents the tail entity, and 'r' signifies the relationship directed from 'h' to 't'. For example, let us consider the triplet (Leonardo da Vinci, painted, Mona Lisa). In this instance, 'Leonardo da Vinci' serves as the head entity, 'Mona Lisa' acts as the tail entity and 'painted' denotes the relationship directed from 'Leonardo da Vinci' to 'Mona Lisa'.

In recent years, knowledge graphs have gained significant prominence as a powerful tool for organizing and using information across various domains. Their graph structure enables the exploration of complex relationships between entities, particularly for natural language

processing (NLP). As a result, knowledge graphs have found applications in diverse areas, showcasing their versatility. Notably, they are employed in voice assisted question-and-answer systems, anomaly detection, recommendation systems, as well as cyber-security and fraud detection [80].

Moreover, knowledge graphs have utility beyond the realms of AI and machine learning. For instance, in healthcare and biomedical research, knowledge graphs play a crucial role in integrating and analyzing complex biological data, leading to advancements in precision medicine, drug discovery, and personalized healthcare [80].

Additionally, knowledge graphs are employed for fact checking to systematically validate claims by cross-referencing structured relationships and information [81]. These structured representations enable fact checkers to uncover patterns and inconsistencies, aiding in the identification of misinformation.

In this chapter, we present our model which uses knowledge graphs for predicting the veracity of claims. In our model, two knowledge graphs are created: one from true claims, referred to as true claims knowledge graph (TCKG), and the other from false claims, referred to as false claims knowledge graph (FCKG). After which, two knowledge graph embedding models are trained on the respective knowledge graphs, using ConvE, a convolutional neural network-based knowledge graph embedding. These two embedding models are merged in parallel to form a combined ConvE model (CCM). This composite model is subsequently employed for predicting the veracity of claims. The evaluation results show that our model achieves high accuracy compared to existing models for claim verification.

Furthermore, in the combined ConvE model, we employ a color-coded scheme to elucidate both true and false triplets, accompanied by the model's confidence score. This visualization aids users in comprehending the rationale behind the model's predictions, offering a visual distinction between the corroborating and opposing triplets derived from the underlying knowledge graphs.

3.1.1 Problem Statement

Given a claim C, we extract triplets (subject, object, and the relation between them) from the claim and classify the claim as true or false and provide explanations. Specifically, a claim C can be expressed as a set of triplets

$$T = \{\ t1,\ t2,\ t3, \ldots,\ tn\ \} \tag{3.1}$$

Given T, we define our problem as a classification task. The classifier cf learns from labeled claims C_i and binary labels Y_i, corresponding to true or false claims,

$$cf \coloneqq C_i \Rightarrow Y_i \tag{3.2}$$

And predicts annotation for a new claim,

$$cf\,(Cj) \rightarrow P_j, reasons_j = \{exp_{t1}, .., exp_{tn}\} \tag{3.3}$$

where P_j is the predicted class label and $reasons_j$ is the set of human understandable explanations in the form of color-coded triplets.

3.2 Knowledge Graph

For a given set of entities E and set of relations R, we consider a knowledge graph $K \subseteq E \times R \times E$ as a directed, multi-relational graph that comprises triplets $(h, r, t) \in K$ in which $h, t \in E$ represent a triplets' respective head and tail entities and $r \in R$ represents its relationship. Figure 3.1 depicts an exemplary knowledge graph. The direction of a relationship indicates the roles of the entities, i.e., head or tail entity. For instance, in the triplet ('Bill Gates', 'Founded', 'Microsoft'), 'Bill Gates' is the head and 'Microsoft' is the tail entity. In a knowledge graphs, the nodes represent entities and edges their respective relations.

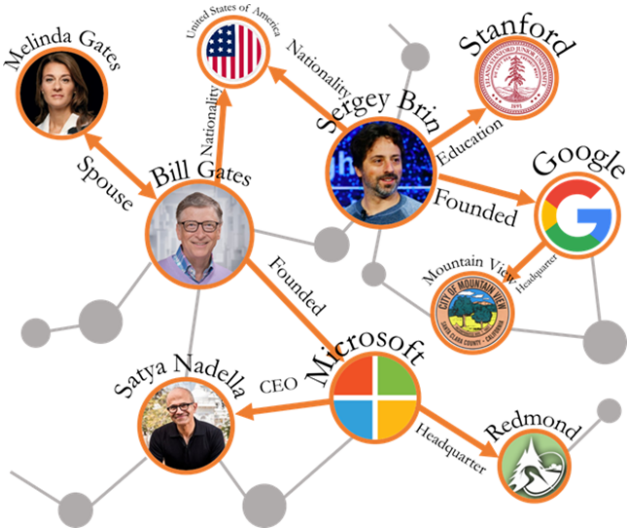


Figure 3.1: Exemplary knowledge graph

In contrast to triplets in a KG, there are different philosophies, or assumptions, for the consideration of triplets not contained in a KG. Under the closed world assumption (CWA), all triplets that are not part of a KG are considered as false. Based on the example in Figure 3.1, the triplet('Bill Gates', 'Founded', 'Goolge') is a false triplet under the CWA since it is not part of the KG. Under the open world assumption (OWA), it is considered unknown as to whether triplets that are not part of the KG are true or false. The construction of KGs under the principles of the semantic web (and RDF) rely on the OWA. While constructing the knowledge graphs for this study we considered the OWA and used negative sampling techniques during the training of knowledge graph embedding models to avoid over-generalization. Negative sampling is defined as the process of introducing false or non-existent relationships as negative

samples used during the training of knowledge graph (KG) embedding models. This technique helps the model differentiate between true and false facts. By altering either the subject or the object in a true relationship, negative samples are created, providing the model with examples of what it should not predict. This approach is vital in sparse KGs, where negative samples are scarce, aiding the model in learning accurate embeddings.

3.2.1 Knowledge Graph Embeddings

Knowledge graph embeddings (KGEs) are techniques used to represent entities $e \in E$ and relationships $r \in R$ in a knowledge graph as dense, low-dimensional vectors. These vectors capture the semantic meaning and relational information of entities and relationships while preserving its structural properties [82], [83]. Given a collection of triplets $T = (head, relation, tail)$, the knowledge graph embedding model produces, for each entity and relation present in the knowledge graph, a continuous vector representation (em_h, em_r, em_t) corresponding to the embedding of a triplet. Where $em_h, em_t \in \mathcal{R}^d$, $em_r \in \mathcal{R}^d$, and d is the embedding dimension. Figure 3.2 shows an embedding of the entities and relations in $\mathcal{R}^2$ from the KG from Figure 3.1.

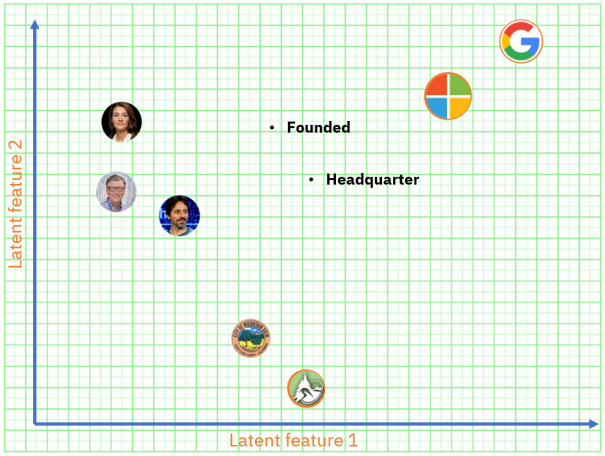


Figure 3.2: Example embedding from the KG portrayed by Figure 3.1

Knowledge graph embeddings have gained significant attention in recent years due to their ability to encode complex relationships in a continuous vector space. By mapping entities and relationships to vectors, knowledge graph embeddings facilitate various tasks such as link prediction, entity classification as well as for downstream tasks such as question answering, recommendation systems and relation extraction [73], [83]. For instance, knowledge graph embeddings enhance the prediction performance of recommendation systems by modeling user-item interactions and capturing item-item relationships [80]. Additionally, knowledge graph embeddings contribute to knowledge graph completion, where missing or erroneous triplets in a knowledge graph are inferred or corrected [73].

One common approach for generating knowledge graph embeddings is through translational models like TransE [43] or TransR [84]. Translational models aim to learn vector representations that capture the translation or transformation between entities and relationships in a knowledge graph. While translational models have shown success in certain scenarios, they have limitations. These models assume that the translation between entities and relationships can be represented by a single vector, which may not always capture the complexity of the relationships accurately [9, 82]. Translational models also struggle with modeling many-to-many relationships and capturing more nuanced semantic meanings of relations [43, 9, 82].

On the other hand, neural network-based knowledge graph embedding models offer unique advantages in representing knowledge graph embeddings, as they can capture both the structural and semantic features of the entities and relations in a knowledge graph [85]. For instance, CNN based models like ConvE [86] are well-suited for capturing local patterns and identifying relevant features within the input data. They can effectively capture the structural and contextual information present in knowledge graphs by using the hierarchical nature of the CNN architecture. CNN models excel at learning complex relationships and dependencies between entities and relations, enabling comprehensive embeddings [87].

3.2.2 Overview of ConvE

ConvE (convolutional 2D knowledge graph embedding) is a popular model in the field of knowledge graph embedding that uses convolutional neural networks (CNNs) to learn embeddings. Introduced by Dettmers [86], ConvE combines the strengths of CNN and knowledge graph embeddings to achieve competitive performance across various knowledge graph-related tasks, including link prediction and entity classification. The core idea revolves around using 2D convolutional layers to capture local patterns and interactions among entities and relations within a knowledge graph.

ConvE represents entities and relations in a knowledge graph using one-dimensional embeddings of size d. This compact embedding representation significantly reduces memory requirements and enables efficient processing of large-scale knowledge graphs. ConvE's architecture consists of three main components: an input layer, a convolutional layer, and a fully connected layer. The input layer reshapes the embeddings of the head entity, relation, and tail entity into two-dimensional matrices. The convolutional layer applies multiple filters to these matrices, capturing local patterns and interactions. The resulting feature maps are then flattened and fed into a fully connected layer, which generates the final embeddings.

The input to ConvE is a matrix $A \in \mathcal{R}^{2\times d}$ where the first row of A represents $h \in \mathcal{R}^d$ and the second row represents $r \in \mathcal{R}^d$. A is reshaped to a matrix $B \in \mathcal{R}^{m\times n}$ where the first $m/2$ half rows represent h and the remaining $m/2$ half rows represent r. In the convolution layer, a set of 2-dimensional convolutional filters $\Omega = \{\omega_i | \omega_i \in \mathcal{R}^{r\times c}\}$ are applied on B that capture interactions between h and r. The resulting feature maps are reshaped and concatenated in order to create a feature vector $v \in \mathcal{R}^{|\Omega|rc}$. In the next step, v is mapped into the entity space using a linear transformation $W \in \mathcal{R}^{|\Omega|rc\times d}$, that is $e_{h,r} = v^T W$. The plausibility of a

triplet $(h, r, t) \in K$ is then given by:

$$f(h, r, t) = e_{h,r}^T t. \tag{23}$$

Since the interaction model can be decomposed into $f(h, r, t) = h^T f_0(h, r), t_i$, the model is particularly designed for 1-N scoring, i.e., efficient computation of scores for (h, r, t) for fixed h, r and many different t [82].

ConvE strikes a favorable balance between embedding quality and computational complexity. By using significantly fewer parameters compared to fully connected neural networks, ConvE is well-suited for resource-constrained environments and handling large knowledge graphs. This is particularly advantageous when dealing with limited computational resources or processing large-scale datasets [86].

3.2.3 Knowledge Graph Embedding Assumptions

As most KGs are constructed using open world assumption (OWA) and contain only positive examples, we require training approaches involving negative sampling techniques to avoid over-generalization to true facts. During training, one of the following assumptions are made for generating the negative triplets: closed world assumption (CWA), open world assumption (OWA), local closed world assumption (LCWA) or the stochastic local closed world assumption (SLCWA).

Open World Assumption

When training under the open world assumption (OWA), all triplets that are not part of the knowledge graph are considered unknown (e.g., neither positive nor negative). This leads to under-fitting (i.e., over-generalization) and is therefore usually a poor choice for training knowledge graph embedding models [88].

Closed World Assumption

When training under the close world assumption (CWA), all triplets that are not part of the knowledge graph are considered as negative. As most knowledge graphs are inherently incomplete, this leads to over-fitting and is therefore usually a poor choice for training knowledge graph embedding models.

Local Closed World Assumption

When training under the local closed world assumption (LCWA; introduced in [89]), a particular subset of triplets that are not part of the knowledge graph are considered as negative. In this setting, for any triplet$(h, r, t) \in K$ that has been observed, a set $T^-(h, r)$ of negative examples is created by considering all triplets $(h, r, t_i) \notin K$ as false. Similarly, we can construct $H^-(r, t)$, using head generation strategy based on all triplets $(h_i, r, t) \notin K$, or $R^-(h, t)$ using relation generation strategy based on the triplets $(h, r_i, t) \notin K$. However,

most of the works in knowledge graph modeling construct only $T^-(h, r)$ using tail generation strategy as the set of negative examples [82].

Stochastic Local Closed World Assumption

When training under the stochastic local closed world assumption (SLCWA), a random subset of the union of the three generation strategies from LCWA are considered as negative triplets. Thus, Instead of considering all possible triplets $(h, r, t_i) \notin K$, $(h_i, r, t) \notin K$ or $(h, r_i, t) \notin K$ as false, we randomly take samples of these sets. There are a few benefits from doing this [90]:

- Reduce computational workload: In knowledge graph modeling, especially when dealing with large datasets, considering all possible negative triplets (i.e., triplets not in the KG) can be computationally expensive. SLCWA significantly lowers the processing requirements by randomly sampling negative triplets instead of considering all possible ones.
- Spare updates: In embedding-based models, each entity and relation in the knowledge graph is represented as a vector in a high-dimensional space. When using SLCWA, only a subset of these embeddings will be updated during each training iteration because only a sample of the negative triplets is considered. This selective updating spares computational resources and can lead to faster convergence, as the model does not need to adjust the embeddings for the entire graph with each iteration.

Based on the above mentioned advantages, we employed the stochastic local closed world assumption (SLCWA) as our primary training approach for the knowledge graph embedding models.

3.3 Methodology

In this section, we describe the methodology of the proposed model in detail. We present the steps used for creation and storage of the knowledge graphs, the classification models used and the training setup of the models.

3.3.1 Knowledge Graph Creation

For creating the knowledge graphs, we use true and false claims seperately. From the true claims we create the true claims knowledge graph (TCKG) which is later used to train the true claims ConvE model (TCCM), whereas the false claims are used to create the false claims knowledge graph (FCKG) which is later used to train the false claims ConvE model (FCCM), refer Figure 3.3

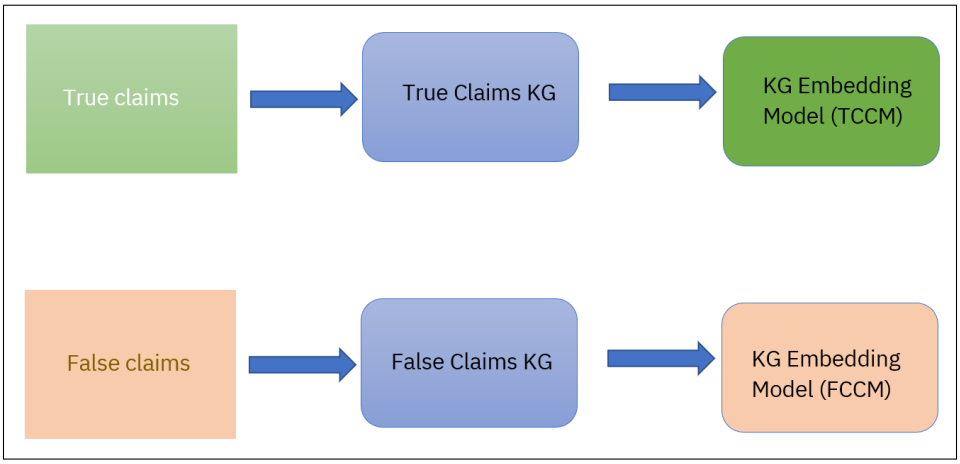


Figure 3.3: Knowledge graphs and models creation process

For creating the knowledge graphs, we first pre-process the claims, then we extract triplets. We use these triplets to create the knowledge graphs. The following sections details the individual processes.

Data Pre-processing

Pre-processing is an essential step in building a knowledge graph from a corpus, as it helps to transform raw text data into a structured and standardized format that is suitable for knowledge representation and extraction. The following are the pre-processing steps involved,

1. *Tokenization*: Split each sentence into individual words or tokens.
    - Example: Input sentence: "I love to play soccer." Tokenized output: ["I", "love", "to", "play", "soccer."]
2. *Co-reference resolution*: Resolve co-references in the text to determine which pronouns or noun phrases refer to the same entity. This step helps in establishing consistent references and connections between entities.
    - Example: Input sentence: "John went to the store. He bought some groceries." Co-reference resolution output: "John went to the store. John bought some groceries."
3. *Lemmatization*: Convert each token to its base or dictionary form (lemma) to normalize the words. This step reduces the inflectional forms of words to their common base form, which helps in entity matching and reducing redundancy.
    - Example: Input tokens: ["running", 'jumps", "jumped"] Lemmatized output: ["run", "jump", "jump"]

4. *Stop word removal*: Remove common words, known as stop words, that do not carry significant semantic meaning and are not useful for extracting information. Examples of stop words include "and", "the", "is" etc.

    - Example: Input sentence: "The cat is sitting on the mat." Stop word removal output: ["cat sitting on mat"]

Triplet Extraction

For triplet extractions from the pre-processed data, we use the following steps:

1. *Named entity recognition (NER)*: NER is a crucial task that identifies mentions of named entities in each text, such as people, organizations, locations, and other types. The entities recognized by NER serve two main purposes within the knowledge graph construction process. Firstly, they can be used to generate new candidate nodes, allowing the incorporation of the identified entities into the graph. Secondly, these entities can be linked to existing nodes through the named entity linking task, establishing connections and enriching the knowledge graph.

2. *Named entity linking (NEL)*: NEL associates mentions of entities in a text with the existing nodes of a target knowledge graph. This task involves resolving and disambiguating entity mentions and linking them to their respective entities in a knowledge base. Additionally, as part of the NEL, entity disambiguation (ED) is also performed to determine the correct entity for each mention. To do this, we use the Wikification process, where Wikipedia is used as a knowledge base to disambiguate entities. However, due to the potential presence of multiple entities in Wikipedia with similar titles, we include both the Wikipedia entity title and their unique ID in the resulting linkage. In cases where a Wikipedia entry is not available, such as for very recent entities, a randomy generated unique ID is provided to represent the entity in the knowledge graph.

3. *Relation extraction (RE)*: RE identifies and extracts relationships between entities mentioned in the text. This task involves two subtasks, parts-of-speech (POS) tagging and dependency parsing. POS tagging assigns a grammatical category (e.g., noun, verb, adjective) to each word or token, providing contextual information about their functions within a sentence. Dependency parsing analyzes the sentence's grammatical structure, revealing the relationships and syntactic dependencies between words. By combining the outputs of POS tagging and dependency parsing, the relation extraction process accurately identifies and extract meaningful relationships between entities.

4. *Joint entity and relation extraction*: Triplet extraction is a challenging task that requires dealing with complex linguistic phenomena, such as long-distance dependencies and syntactic variations. Along with the previous mentioned steps, we use REBEL [91], an end-to-end sequence-to-sequence model that generates relation triplets from raw

text, without relying on predefined relation schemas or hand-crafted features. REBEL is based on BART, a pre-trained language model that can both encode and decode natural language. REBEL can handle more than 200 different relation types and has been pre-trained on various RE benchmarks, including a large-scale distantly supervised dataset curated by the authors of REBEL. We merge the triplets extracted earlier with those obtained using REBEL, retaining only the distinct triplets.

Regular Updating of Knowledge Graphs

Regular updating of the knowledge graphs is essential to ensure their relevance over time. By continuously incorporating new information, the knowledge graphs remain up to date with the latest knowledge. To acquire new knowledge, we collect both true and false claims from three different fact-checking websites, namely "Polygraph.info", "Politfact.com" and "Snopes.com". We use the previously mentioned pre-processing steps and the triplet generation pipeline for updating the knowledge graphs, refer Figure 3.4.

- "Polygraph.info" [1] is a fact-checking website that focuses on debunking false or misleading claims with a particular emphasis on disinformation related to global affairs.
- "Politifact.com" [2] is a highly reputable fact-checking platform known for its thorough evaluation and rating of the accuracy of various claims, ensuring that readers have access to reliable information and aiding in the fight against misinformation.
- "Snopes.com" [3] is a well-known website that verifies and debunks urban legends, rumors, and various misinformation circulating online.

[1] www.polygraph.info
[2] www.politifact.com
[3] www.snopes.com

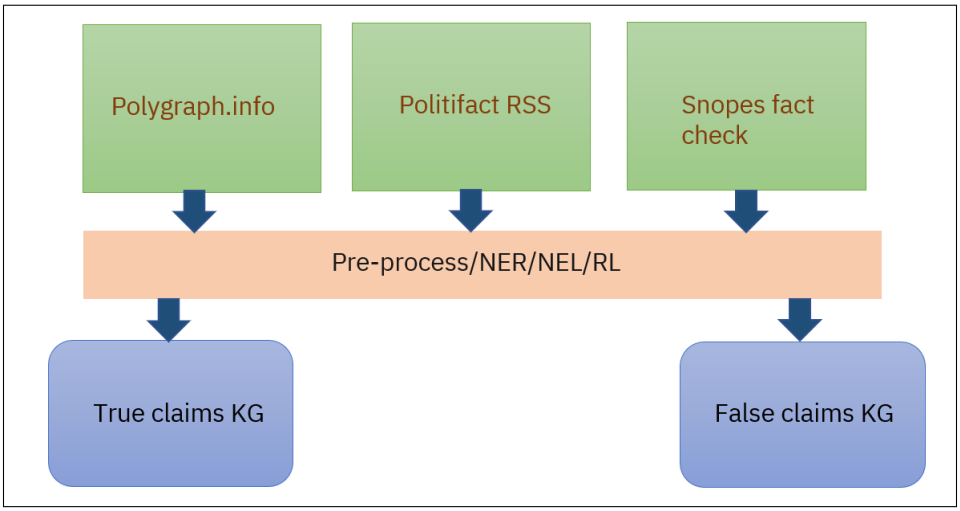


Figure 3.4: Addition on current knowledge to the knowledge graphs

Storage of Knowledge Graphs

Storage of knowledge graphs refers to the methods and technologies used to store and manage the structured data that constitutes a knowledge graph.

To store knowledge graphs effectively, the most popular approaches are as follows,

- Triplestores: Triplestores are specialized databases designed explicitly for storing and querying RDF (resource description framework) triplets, which consist of head-relation-tail statements. Triplestores offer efficient storage and retrieval of semantic data and are commonly used for knowledge graphs.
- Graph databases: Another option for storing knowledge graph is through graph databases. These databases organize data as nodes, edges, and properties, allowing for efficient representation and traversal of complex graphical relationships. Examples of graph databases include Neo4j, Amazon Neptune, and JanusGraph.
- Relational databases: Traditional relational databases can also be used for storing knowledge graphs by mapping the graph structure to tables and relationships. Popular relational databases like PostgreSQL, MySQL, and Oracle Database can be employed for this purpose.

In a comparative study conducted by Sikhinam [92], the performance of a graph database (Neo4j) and a relational database (PostgreSQL) was evaluated using TPC-H [93], a decision support benchmark. The study concluded that loading and reading data was faster in the relational database compared to the graph database. Additionally, the graph database occupied more disk space for storing the same amount of data. Finally, when the database schema remains constant, the study suggested that a relational database is a better option.

Considering that the properties of nodes and relations in our study remain unchanged, and a constant schema is desirable, PostgreSQL was used to store the knowledge graphs. PostgreSQL is a powerful open-source relational database management system known for its reliability, extensibility, and support for advanced features such full-text search, and spatial data [94].

3.3.2 Overall Architecture of the Proposed Model

Figure 3.5 depicts the overall architecture of our proposed model. Given a new claim C, we follow a series of steps to process the claim and make a prediction. The steps are described as follows:

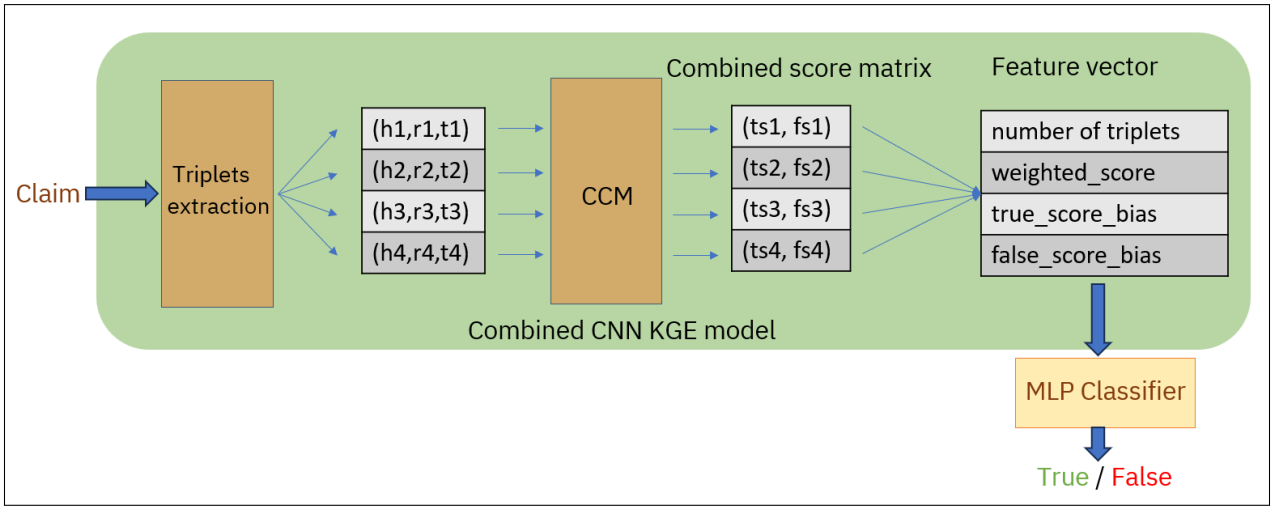


Figure 3.5: The overall architecture of the KG-FC model

1. *Triplets extraction:* Firstly, we extract triplets from the claim. Each triplet is represented as (h, r, t), where 'h' represents the head entity or subject, 'r' represents the relation or predicate, and 't' represents the tail entity or object. These triplets capture the key elements and relationships mentioned in the claim.

2. *Combined ConvE model:* These extracted triplets vector is then passed through the combined ConvE model (CCM) to produce the 2D score matrix. Where each record in the matrix denotes the scores of the corresponding triplet. For a given triplet (h,r,t), the score is represented as (ts, fs), where ts is the score generated by the true claims ConvE model (TCCM) and fs is the score generated by the false claims ConvE model (FCCM). The scores in the matrix are determined by the CCM's ability to learn the underlying patterns and relationships between the entities and relations, enabling it to evaluate the plausibility of each triplet based on the given claim. The length of

the score matrix is equal to the number of triplets in the claim. Although there are no limitations on the number of triplets in a claim, they are typically fewer than the number of triplets found in a news article.

3. *Feature vector generation:* From the 2D score matrix obtained from the CCM, we derive a 1D feature vector. This feature vector serves as a condensed representation of the extracted triplets and their respective scores. It captures the important information necessary for making predictions.

4. *Classifier:* Finally, we pass the feature vector to a trained multi-layer perceptron (MLP) classifier. The MLP classifier is a supervised learning algorithm used for classification tasks. It takes the feature vector as input and predicts the final outcome or label for the given claim. The MLP classifier has been trained on labeled data to learn the patterns and characteristics of true and false claims.

The overall process involves extracting triplets from the claim, representing them as triplets, passing them through the ConvE model to obtain a score matrix, generating a feature vector from the score matrix, and then using a MLP classifier to make the final prediction. Each step in this process contributes to the overall prediction performance of our proposed model.

The following sections will provide detailed descriptions of each individual module in our model, shedding light on their inner workings and the techniques.

3.3.3 Combined ConvE Model

We build the combined ConvE model (CCM) using the combination of the true claims ConvE model (TCCM) and the false claims ConvE model (FCCM). CCM incorporates TCCM and FCCM to improve the prediction performance of our fact checking model. By training these models separately on the true claims knowledge graph (TCKG) and the false claims knowledge graph (FCKG) respectively, we use the strengths of both models and achieve higher accuracy, precision, and recall.

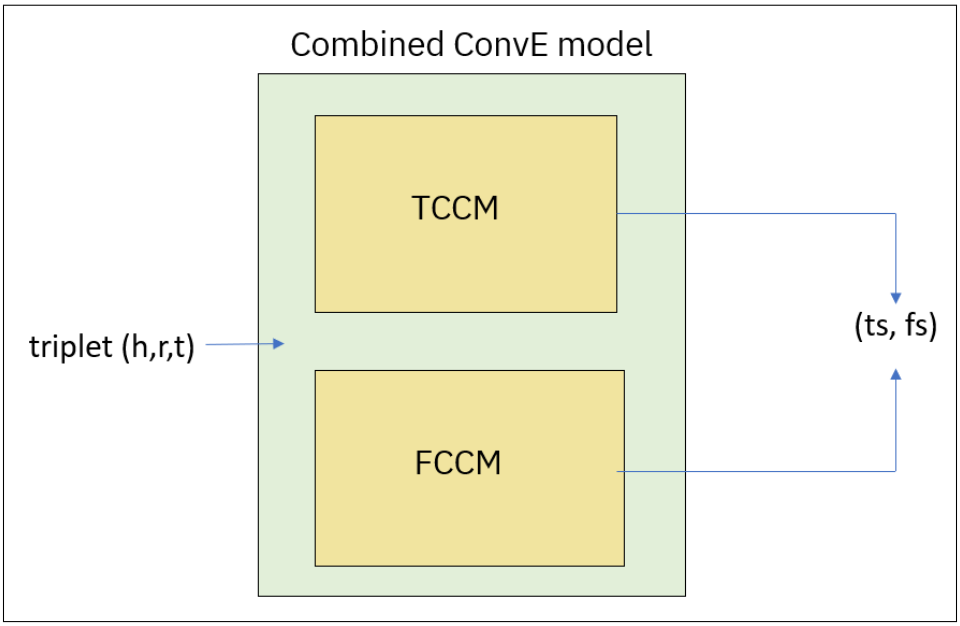


Figure 3.6: Combined ConvE model for one triplet (h,r,t), where ts and fs are the scores calculated for the triplet from TCCM and FCCM respectively.

Figure 3.6 provides a visual representation of the setup, where TCCM represents the ConvE model trained on TCKG, and FCCM represents the ConvE model trained on FCKG. These models are trained independently to capture the characteristics and patterns specific to each knowledge graph. We then use TCCM and FCCM in parallel as shown in the Figure 3.6 to get the triplet scores (ts, fs) independently from the models.

The combined ConvE model (CCM) takes advantage of the complementary information learned from TCCM and FCCM. When presented with a new statement to fact check, the CCM uses the knowledge learned from TCCM and FCCM in parallel. By integrating the true claims ConvE model (TCCM) and the false claims ConvE model (FCCM), the CCM discerns the veracity of statements or claims.

3.3.4 Score Generation for One Triplet

The high level ConvE architecture consists of an encoding component and a scoring component. Given an input triplet (h, r, t), the encoding component maps entities $h, t \in E$ to their distributed embedding representations $e_h, e_t \in R^d$, with d being the embedding dimension.

In the scoring component, the two entity embeddings e_h and e_t are scored by a function ψ_r. The score of a triplet(h, r, t) is defined as $\psi(h, r, t) = \psi_r(e_h, e_t) \in R^d$.

Formally, the scoring function is defined as Equation 3.4:

$$\psi_r(e_h, e_t) = f(vec(f([\bar{e_h}, \bar{r_r}] * \omega))\mathbf{W})e_t \tag{3.4}$$

where $r_r \in R^d$ is a relation parameter depending on r, $\bar{e_h}$ and $\bar{r_r}$ denote a 2D reshaping of e_h and r_r, respectively: if $e_h, r_r \in R^d$, then $\bar{e_h}, \bar{r_r} \in R^{d_w \times d_h}$, where $d = d_w \times d_h$.

In the feed-forward pass, the model performs a row-vector look-up operation on two embedding matrices, one for entities denoted $\mathbf{E}^{\varepsilon \times d}$, and one for relations denoted $\mathbf{R}^{R \times d'}$, where d and d' are the entity and relation embedding dimensions, and $|\varepsilon|$ and $|R|$ denote the number of entities and relations.

The model then concatenates e_h and r_r, and uses it as an input for a 2D convolutional layer with filters ω. Such a layer returns a feature map tensor $\tau \in R^{c \times m \times n}$, where c is the number of 2D feature maps with dimensions m and n. The tensor τ is then reshaped into a vector $vec(\tau) \in R^{c \times m \times n}$, which is then projected into a d-dimensional space using a linear transformation parameterised by the matrix $\mathbf{W} \in \mathbf{R}^{c \times m \times n \times d}$ and matched with the object embedding e_t via an inner product. The parameters of the convolutional filters and the matrix $\mathbf{W}$ are independent of the parameters for the entities h and t and the relationship r. After which we apply the sigmoid function, $\sigma(\cdot)$ to the scores, that is $p = \sigma(\psi_r(e_h, e_t))$

Figure 3.7 shows the visualization of the separate steps for scoring a triplet.

1. Embedding matrices: The model has two embedding matrices: $\mathbf{E}$ for entities and $\mathbf{R}$ for relations. These matrices have dimensions $\mathbf{E} \times d$ and $\mathbf{R} \times d'$, respectively. Here, d represents the entity embedding dimension, and d' represents the relation embedding dimension.

2. Look-up operation: During the feed-forward pass, the model performs a row-vector look-up operation on the embedding matrices. Given a triplet (head entity, relation, tail entity), the model retrieves the corresponding row vectors from the entity embedding matrix $\mathbf{E}$ and the relation embedding matrix $\mathbf{R}$. Let's denote the head entity row vector as e_h and the relation row vector as r_r.

3. Concatenation: The model concatenates the head entity row vector e_h and the relation row vector r_r to create a combined input vector. Combined vector is denoted as $[e_h, r_r]$.

4. Convolutional layer: The combined input vector $[e_h, r_r]$ is passed through a 2D convolutional layer with filters ω. The convolutional layer applies the filters to capture local patterns and relationships in the input. This operation produces a feature map tensor.

5. Reshaping: The feature map tensor τ is reshaped into a vector $vec(\tau)$. This reshaping operation converts the 3D tensor into a 1D vector while preserving the order and arrangement of the values.

6. Linear transformation: The reshaped vector $vec(\tau)$ is projected into a d-dimensional space using a linear transformation. This linear transformation applies a matrix multiplication operation to map the vector into the d-dimensional space.

7. Inner product: The transformed vector is then matched with the object embedding e_t using an inner product operation. The inner product is calculated between the

transformed vector and the object embedding e_t, resulting in a score that represents the model's confidence or compatibility for the given triplet (head entity, relation, tail entity).

8. Sigmoid function: The sigmoid function is applied to the result to produce a score between 0 and 1.

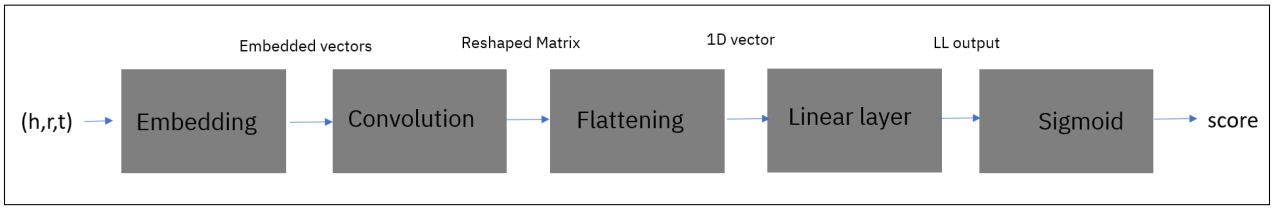


Figure 3.7: Score calculation for a triplet (h,r,t)

It is important to note that the parameters for the convolutional filters and the matrix **W** are independent of the parameters for the entities and relations. This means that they are learned separately during the training process and do not directly depend on the entity and relation embeddings.

To generate the combined score, the CCM first passes the triplet through TCCM and FCCM independently, as shown in Figure 3.6. Each model calculates a score based on its learned embeddings, capturing the relationship between the head entity, relation, and tail entity in the triplet. These scores represent the models' confidence for the likelihood of the triplet, with higher scores indicating a higher likelihood of the triplet being true or false according to each model's perspective.

Next, scores from TCCM and FCCM are passed through an aggregator function, which produces a tuple (ts, fs), where ts represents the score from TCCM and fs represents the score from FCCM.

3.3.5 Score Generation for a Claim

Each claim may consists of multiple triplets. A given claim C can be expressed as a set of triplets $T = \{ t1, t2, t3, \ldots, tn \}$. Each of the triplets (t) are passed through the scoring module to produce the score of a claim (S), as shown in Equation 3.5.

$$S = \{\text{score}(t) : t \in T\} \tag{3.5}$$

Here, S is the 2D matrix of scores, and score(t) denotes the score generated for each individual triplet t within the claim as shown in Figure 3.8.

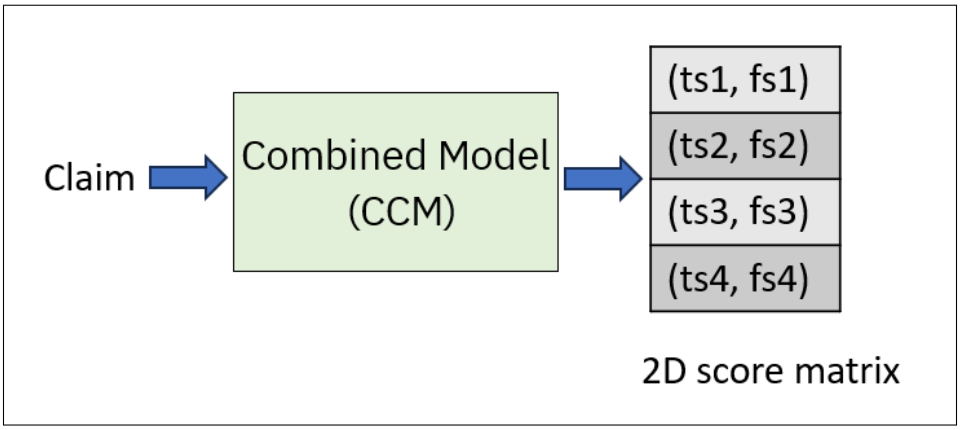


Figure 3.8: Combined score calculation of a claim

3.3.6 Feature Vector Extraction

From the 2D score matrix we extract feature vector $\mathbf{V}$, the input to the classifier.

1. Number of Triplets

This denotes the number of triplets present in the claim.

$$num_{triplets} = |[S]| \tag{3.6}$$

Where, S is the 2D score matrix.

2. Weighted Combined Score

This feature represents the degree of veracity of the triplets within the claim. Additionally, the weight component factors in the importance of a subject in the triplet relative to the claim.

Consider a triplet (h, r, t). The combined score (cs) is first derived by:

$$cs = |ts - fs| \tag{3.7}$$

Where (ts, fs) represents the tuple of scores of the triplet (h, r, t) obtained from the combined model CCM. Let W represent the frequency of the head entity of the triplet relative to the claim, such that $W \in \{|e_h|, |e_t|\}$. The vectors $\mathbf{W}$ and CS consist of the weights (W) and combined scores (cs) of all the triplets in the claim, i.e., $\mathbf{W} = [W]$ and $CS = [cs]$. Here, $|W| = |CS| = num_{triplets}$.

The weighted combined score (WCS) is computed as the dot product of $\mathbf{W}$ and CS, as depicted by the formulae:

$$WCS = \mathbf{W} \cdot CS \tag{3.8}$$

3. True Score Bias

This is defined as follows,

$$T_{bias} = max(|ts_i - ts_j|) \tag{3.9}$$

Where $i \neq j$ and $1 \leq i, j \leq \text{num}_{\text{triplets}}$.

This represents the maximum score difference among all the triplets in respect to the true claims ConvE model (TCCM). This is used to denote the degree of variation of truthfulness of the triplets.

4. False Score Bias

This is defined as follows,

$$F_{bias} = max(|fs_i - fs_j|) \tag{3.10}$$

Where $i \neq j$ and $1 \leq i, j \leq \text{num}_{\text{triplets}}$.

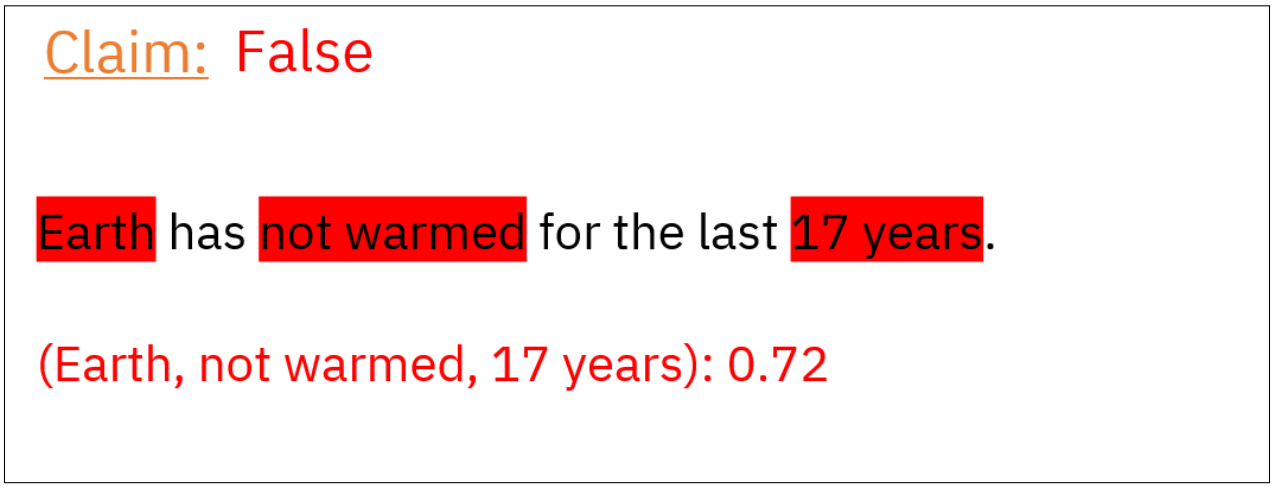
This represents the maximum score difference among all the triplets in respect to the false claims ConvE model (FCCM). This is used to denote the degree of variation of falsehood of the triplets.

3.3.7 Classification of Claims

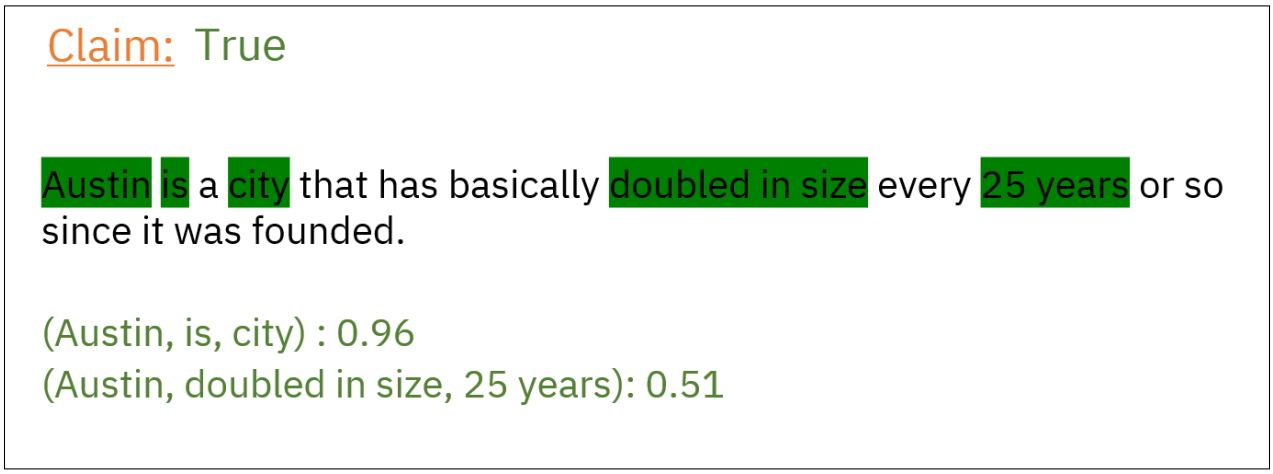
The feature vector is then passed through a multi-layer perceptron (MLP) classifier, which predicts the veracity of the claim. The MLP classifier is a feed-forward neural network used in supervised learning for classification tasks. It comprises three or more layers of nodes: an input layer, one or more hidden layers, and an output layer. MLP is capable of modeling complex, non-linear relationships in data and is often utilized in a variety of fields including image and speech recognition, along with numerous other classification problems. Furthermore, MLP can handle large datasets efficiently and are robust to overfitting, making them valuable tools for both research and practical applications in the field of machine learning. In this study we use a MLP classifier with two hidden layers with (5,2) nodes respectively.

3.4 Explainability of the Model

Given a claim, *"World-renowned singer Taylor Swift's collaboration with GlobalTech has resulted in the development of a new medicine to transport humans."*, our model classifies the claim as true or false. Additionally, it points out the false triplets and true triplets in the claim in a colour coded manner and provides the confidence scores for each fact triplet. Figure 4.9 shows a sample outputs of our base model given a claim along with the confidence scores.



(a) False claim



(b) True claim

Figure 3.9: Sample outputs of our base model given a claim

After illustrating how our model evaluates a specific claim and provides explanations, we now transition to the underlying methodology of the explainability. Given a claim, we represent the claim as a set of triplets. $T = \{t1, t2, t3 \ldots, tn\}$. For every triplet (t_i), we find the combined score tuple (ts, fs).

We represent the triplet veracity $(TV_i$ of the triplet (t_i) as,

$$TV_i = \begin{Bmatrix} \text{True}, & tcs > fcs \\ \text{False}, & fcs > tcs \end{Bmatrix} \tag{3.11}$$

And the confidence score (CS_i) of the model for the triplet (t_i) as,

$$CS_i = \begin{Bmatrix} tcs, & tcs > fcs \\ fcs, & fcs > tcs \end{Bmatrix} \tag{3.12}$$

Using the TV_i we render the triplets as true or false in a color-coded scheme and use CS_i to provide the model's confidence score for each triplet.

3.5 Evaluation

In the following sections, we discuss the dataset used for the study. Then we present the metrics used for evaluation. After which, we illustrate the experimental setup and show the classification report of our model. Subsequently, we showcase the results from an ablation study featuring three variants of the base model. Finally, we compare our model with six baseline state-of-the-art misinformation detection models.

3.5.1 Datasets

In this study we use the LIAR dataset. LIAR dataset was created by Wang [95], which is one of the largest fact checking datasets, containing 12,836 claims along with its veracity determined by human evaluators. The dataset is labelled with a discrete set of values from one to six, corresponding to pants-fire referring to completely false, false, mostly-false, half-true, mostly-true, and true. As our problem is based on binary class classification, we transform these into two labels. Pants-fire and false are contemplated as false, mostly-true, and true as true and we disregard mostly-false and half-true.

The dataset is split into 80:20 for training and testing. The training data contains 56% true and 44% false statements, and the testing data contains 56% true and 44% false statements. There is not much imbalance between the different labels, as shown in Figure 3.10.

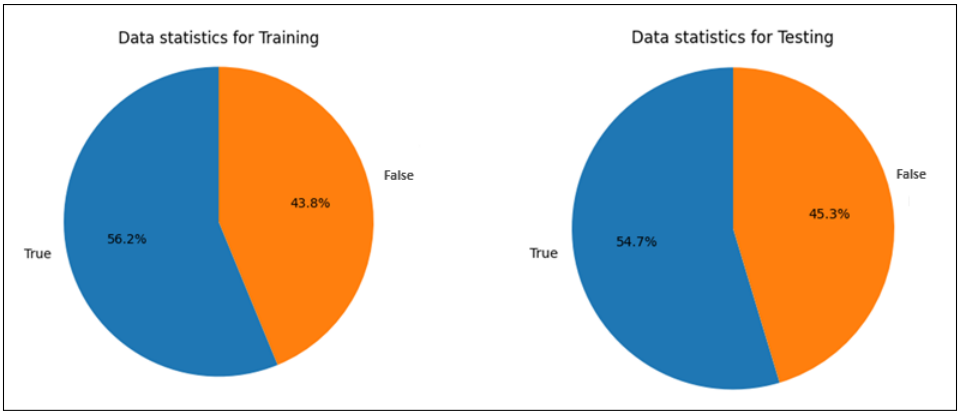


Figure 3.10: LIAR dataset characteristics

Table 3.1 displays sample statements along with each of the label, showing a glimpse of what the dataset looks like.

The dataset provides some additional metadata like the subject, speaker, job, state, party, context, history. However, in the real-life scenario, we may not have this metadata always available. Therefore, we experiment only on the texts of the dataset.

Table 3.1: Sample statements along with each of the label in LIAR dataset

ID	Claim	Label
1	There are already more American jobs in the solar industry than in coal mining.	True
2	Since 1968, more Americans have died from gunfire than died in all the wars of this country's history.	True
3	President Obama has the lowest public approval ratings of any president in modern times	False
4	The Air Force wants taxpayers to fund a fantasy football league.	False

3.5.2 Metrics

Let TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively. We employed four evaluation criteria extensively used in text classification tasks, namely accuracy, precision, recall and F1 score (calculated as in the equations below), to evaluate the performance of the models:

- Accuracy (A): Accuracy is a measure of the classifier's ability to correctly classify a piece of claim as either true or false.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.13}$$

- Precision (P): Precision is a measure for the classifier's exactness such that a low value indicates a large number of false positives. The precision represents the number of positive predictions divided by the total number of positive class values predicted.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.14}$$

- Recall (R): Recall is considered a measure of a classifier's completeness (e.g., a low value of recall indicates many false negatives), where the number of true positives is divided by the number of true positives and the number of false negatives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.15}$$

- F1 score (F1): The F1 score is calculated as the weighted harmonic mean of the precision and recall measures of the classifier.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.16}$$

3.5.3 Experimental Setup

All experiments are conducted on a machine with Intel(R) Core(TM) i7-8750H 2.20GHz CPU, 16 GB of RAM and NVIDIA GeForce RTX 4090 GPU running Windows 11. For the ConvE model, we adhered to the default hyper-parameters as recommended by the authors in [86] and used binary cross entropy loss for training. Table 3.2 details the hyper-parameters used in the study. To maintain consistency and simplicity in our experimental approach, the same hyper-parameter settings were applied to both the true claims ConvE model (TCCM) and false claims ConvE model (FCCM) and trained independently. Additionally, we use the stochastic local closed world assumption (SLCWA) in the training loop for generating the negative triplets.

Table 3.2: Hyper-parameter of the ConvE embedding model

Hyper-parameter	**Value**
Embedding size	200
Batch size	128
Learning rate	0.001
Epochs	1000

3.5.4 Results and Discussion

Table 3.3 depicts the classification report of our model. We can observe from the table that precision is higher for "True" class whereas recall is higher for "False" class. A higher precision value for the "True" class means that the model has a lower tendency to make false positive predictions for true claims. It indicates that when the model predicts the veracity of a claim as true (positive prediction for the "True" class), it is more likely to be correct. In other words, the model is better at avoiding false claims or misinformation while identifying true claim instances. Whereas a higher recall for the "False" class indicates that the model is better at capturing most of the actual false claims present in the dataset, reducing the risk of false negatives for false claims.

Table 3.3: Classification report of the proposed model

	Precision	**Recall**	**F1 score**
Class True	**0.88**	0.86	0.86
Class False	0.84	**0.92**	**0.87**
Macro	0.86	0.89	**0.87**

3.5.5 Ablation Study

To understand the specific impact of each knowledge graph on our model's prediction performance, we conduct ablation experiments. The experimental results are shown in Table

3.4.

The ablation experiments include the following three variants of our model:

- With only true claims knowledge graph (TCKG)
- With only false claims knowledge graph (FCKG)
- With both knowledge graphs

For the first two experiments, the respective scores from FFCM and TFCM are set to zero for each triplet.

Table 3.4: Ablation study results

Our model	**Precision**	**Recall**	**Accuracy**	**F1 Score**
With only False KG	0.78	0.83	0.76	0.80
With only True KG	0.84	0.81	0.81	0.82
With both True KG and False KG	**0.86**	**0.89**	**0.89**	**0.87**

From Table 3.4, we can observe that with the accuracy of with only false claims knowledge graph is the least. This can be attributed to the fact that the knowledge graph constructed from false claims lacks the necessary factual relationships and context, leading to limitations in identifying reliable patterns and distinguishing false claims accurately. Without a comprehensive representation of true claims and their associations, the model's ability to differentiate between reliable and misleading claims is compromised.

On the other hand, the accuracy with only the true knowledge graph is higher, showcasing the importance of a knowledge graph built from verified and reliable claims. The true knowledge graph contains valuable structured information, capturing genuine relationships between entities and their attributes, which aids in the identification of trustworthy patterns in factual claims. Consequently, the model trained on the true knowledge graph demonstrates improved prediction performance in detecting true claims with higher precision and accuracy.

However, the highest accuracy is achieved when the model uses both the true claims knowledge graph and the false claims knowledge graph, with an increase of almost 8% to that of only using true knowledge graph. Integrating knowledge from both sides provides a more comprehensive and balanced representation of information. By combining the knowledge of true and false claims, the model gains a broader understanding of the entire information landscape. This enables the combined model, CCM to effectively leverage contrasting patterns and identify distinctive features between reliable and misleading claims.

3.5.6 Comparison with Baseline Models

We conducted a comparative analysis of our proposed model against six state-of-the-art misinformation classification baseline models, all of which were evaluated using the same LIAR dataset by their authors as documented in their respective publications.

- Khan [96]: Explored several advanced pre-trained language models for misinformation detection along with the traditional and deep learning ones. They found that that BERT and similar pre-trained models perform the best for misinformation detection, especially RoBERTa for the LIAR dataset.
- Jayaraman [97]: XLNet base fine-tuning model for fake news detection. This model uses permutation language modeling to capture the right and left context information jointly and computes contextualized input representation.
- Mehta [98]: In this work BERT is fine-tuned for domain-specific datasets, incorporating human justification and metadata to enhance model performance. The approach involves employing multiple BERT models with shared weights to manage various inputs and use extra justification and metadata.
- Yang [65]: An unsupervised learning framework, UFD, which utilizes a probabilistic graphical model to model the truths of news and the users' credibility. An efficient collapsed Gibbs sampling approach is proposed to solve the inference problem.
- Nida [99]: The authors used a multi-modal deep learning model for misinformation detection. For the textual attributes Bi-LSTM-GRU-dense deep learning model was used, while for the remaining attributes, dense deep learning model was used.
- Bhatt [100]: The authors used an enhance sequence model with four branches having inputs as statement (S branch), metadata (M branch), justification (J branch) and credit score (C branch) for classification. The authors use credibility information and metadata associated with the article.

Table 3.5: Comparison between the proposed and state-of-the-art models

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Khan [96]	0.62	0.63	0.62	0.63
Jayaraman [97]	0.72	0.52	0.39	0.45
Mehta [98]	0.74	0.70	0.85	0.77
Yang [65]	0.76	0.75	0.75	0.75
Nida [99]	**0.89**	**0.87**	0.88	**0.87**
Bhatt [100]	0.82	0.73	0.71	0.72
Our model	**0.89**	0.86	**0.89**	**0.87**

Table 3.5 presents a comparison between our approach and baseline models. The results suggest that the proposed model demonstrates superior prediction performance compared to the majority of state-of-the-art approaches. Notably, our model achieves an accuracy of 89%, which is on par with the best performing model presented by Nida.

Importantly, it should be highlighted that Nida used supplementary features, including author information and content metadata, which might not always be available in real world scenarios. In contrast, our model exclusively relies on textual features of the content.

3.6 Limitations of the Model

Like any approach, our model has certain limitations as follows,

- The accuracy of our model heavily depends on the quality and completeness of the underlying knowledge graphs. Inaccurate or biased claims within these graphs could lead to erroneous predictions. Moreover, the computational resources required for training and inference using convolutional neural networks and knowledge graph embeddings might pose challenges in resource-constrained environments.

- The interpretability provided by XAI techniques might not always cover all aspects of the model's decision making process. For instance, it does not attribute the sources of the information.

- Although our model benefits from periodic updates to the knowledge graphs, it remains constrained by the intervals between updates. Claims emerging between these updates are not immediately validated.

3.7 Chapter Summary

The chapter presents a method for fact checking using CNN-based knowledge graph embeddings. This method involves the creation of two knowledge graphs constructed from verified true and false claims, respectively, which are later used for training two separate knowledge graph embedding models. By employing the ConvE technique, which is based on convolutional neural networks, in conjunction with feature vector generation and classification methods, the model predicts the veracity of claims. Experimental results show that the proposed model outperforms the baseline models.

The output of the KG-based fact checking model is further enhanced by XAI techniques. True and false triplets are highlighted by different colors, green and red, respectively. In addition, the combined ConvE model generates confidence scores for triplets. XAI output allows end users to understand the reasons behind the model's predictions.

For future work, we plan to extend the work in the following directions:

- Exploring alternative types of neural network embedding models to further enhance the accuracy of the fact checking process.

- Evaluating the model's prediction performance across a broader range of datasets, especially those from different domains.

- Investigating further into XAI techniques to provide more detailed and understandable explanations for the model's predictions.

Chapter 4

Fact Checking on Twitter Using Diffusion Dynamics and Knowledge Graphs

In this chapter, we describe the propagation-based model that uses diffusion cascades of tweets on Twitter to determine if a post is likely to be misinformation or not. The propagation-based model is combined with the KG-based fact checking model (Chapter 3) to further improve the performance of misinformation identification.

The chapter is structured as follows: Section 4.1 discusses the importance of misinformation identification on Twitter and the motivation of our work. Section 4.2 describes the propagation structures of a tweet on Twitter. Section 4.3 details the methodology used by our model. In Section 4.4 we discuss the datasets used for the study. Section 4.5 outlines the explainability of the model. Section 4.6 presents the evaluation of our model using different metrics. Section 4.7 lists the limitations of the model. Finally, we summarize the chapter in Section 4.8.

4.1 Introduction

Misinformation and disinformation are characterized as narratives or statements whose accuracy is either misleading or intentionally false. They differ in terms of intent and impact. Misinformation is the unintentional spread of false claims while disinformation is the deliberate spread of false claims with harmful intent [1]. These deceptive claims can be detrimental, as they possess the potential to induce panic among the populace and sow the seeds of social unrest [101].

On the bright side, numerous studies have indicated that individuals tend to abstain from disseminating misleading claims if they are aware of its falseness [102]. However, detecting such falsehood is a complex endeavor that necessitates investigative journalism. Furthermore, with the exponential expansion of online content and the widespread use of social media, distinguishing between truth and falsehood has become increasingly challenging [103].

This leads to the problem of misinformation detection on social media platforms. The detection of misinformation is of paramount importance for ensuring the responsible use of popular platforms like Twitter and Instagram, given their extensive reach. In this chapter,

we concentrate on the verification of claims propagation on Twitter, a platform frequently referred to as the "global town square" [104] where many people turn to for updates on current events.

Typically, content-based fact checking methods take center stage for misinformation identification [15]. However, it is important to recognize that online social networks have additional information, such as comments, user emotions, and propagation patterns, that can be used in conjunction with content-based fact checking methods to increase the accuracy of misinformation identification. Analyzing user comments provides insights into community sentiment and skepticism, helping identify potential falsehoods. User emotions serve as indicators of emotional manipulation tactics used in misinformation. While comments and user emotions offer insightful cues, the analysis of propagation patterns emerges as an influential factor. Studying propagation patterns, allows for the identification of suspicious or abnormal dissemination patterns [20], [21].

In a study funded by Twitter, Vosoughi [48] discovered that misinformation exhibits distinct diffusion patterns compared to true claims. Their findings indicated that misinformation spreads significantly farther, faster, deeper, and wider across various categories. However, a subsequent study by Juul [49] replicated the comparison of the spread of fake and genuine claims using the same dataset. While their results largely supported Vosoughi's initial findings, they attributed the four qualities—further, faster, deeper, broader—to the "infectiousness" of the claims. They used two models, namely the SIR (Susceptible-Infectious-Recovered) and IC (Independent Cascade) models, to demonstrate that the discrepancies in dissemination could be solely explained by differences in "infectiousness". This led them to the conclusion that misinformation is inherently more "infectious" than the truth.

While both studies provide empirical evidence that veracity of posts(tweets) can be differentiated using propagation cascades, it remains unclear how this difference emerges and how it can be properly used to create verifiable automated detection mechanisms. Additionally, researchers have attempted to construct automated fact checkers, relying primarily on the work of Vosoughi without considering the factor of "infectiousness" [50, 20].

In this thesis, we use propagation patterns of tweets on Twitter to enhance the classification accuracy of the knowledge graph-based model described in Chapter 3 for fact checking on Twitter. Given a claim (post) circulating on Twitter, we propose a propagation-based model to classify the veracity of the claim. The proposed model is an ensemble classifier which uses the diffusion pattern of a tweet in Twitter. The ensemble classifier uses the following information for the task: (a) temporal and spatial properties of diffusion cascades of tweets, and (b) the "infectiousness" property of misinformation by using insights learned from the spread of COVID-19 and general epidemiological models of infectious diseases. Then we combine this propagation-based ensemble model with the knowledge graph-based model described in Chapter 3 for multi-modal classification of Twitter posts. For the multi-modal fusion we evaluate two popular fusion strategies: early and late fusion. Early fusion combines features from different modalities before processing, while late fusion combines them at the decision level after separate processing [105].

4.2 Propagation Structure Representations

Propagation networks of claims on online social networks are represented in various ways. For this study we employ the following two representations,

- Hop-based structure
- Time-based structure

4.2.1 Hop-based Structure

In this type of structure, the diffusion of a post is represented as a directed acyclic graph (DAG), with the root of the tree being the source tweet and the corresponding children being the retweets as shown in Figure 4.1.

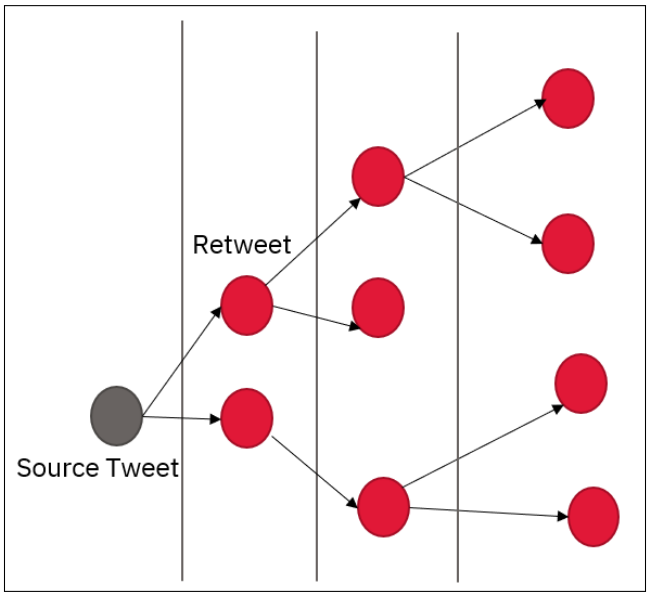


Figure 4.1: Hop-based Representation

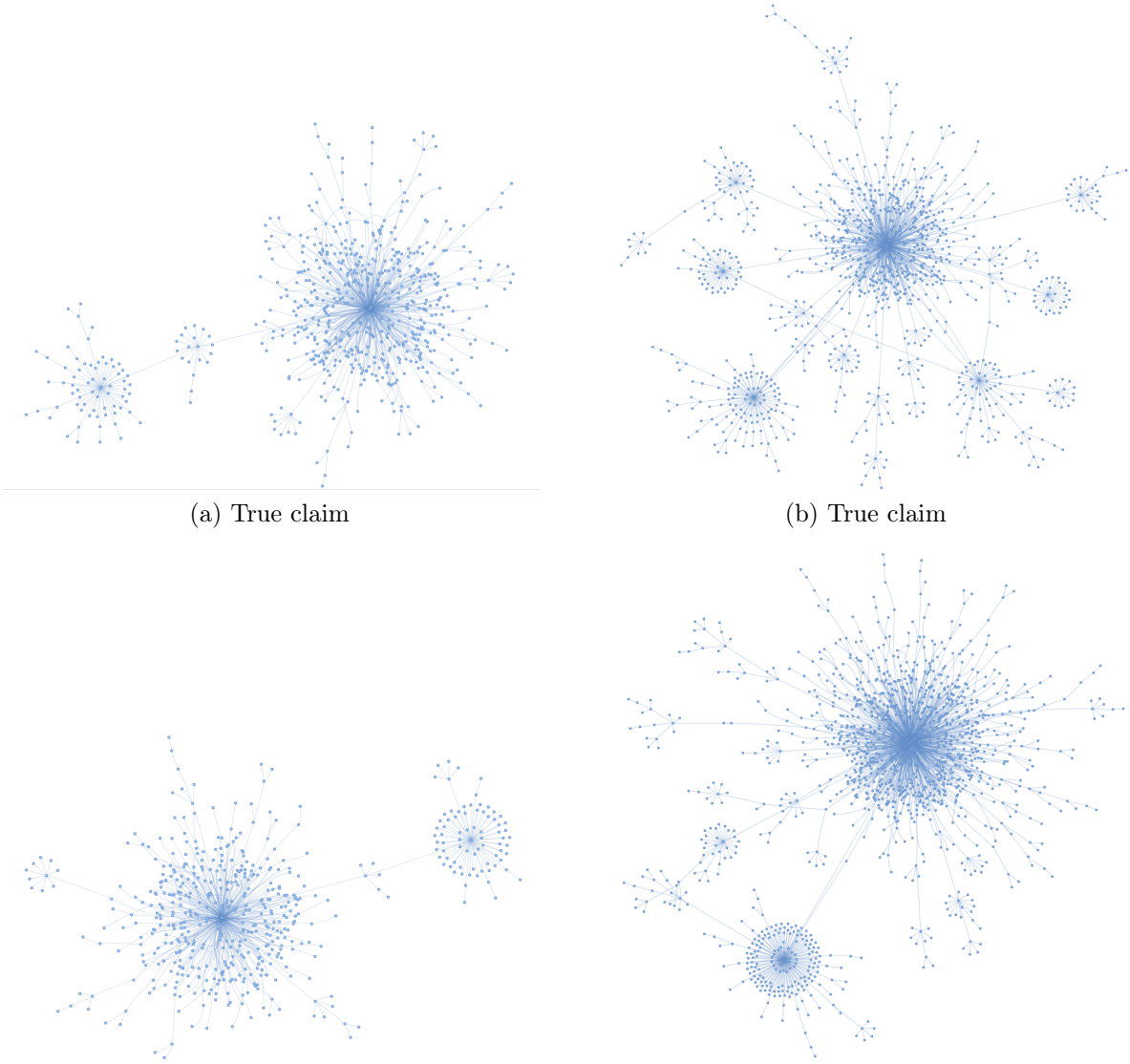
The advantage of using this representation is the ease of using the spatial properties of the diffusion of the social media post. Additionally, this method of representation effectively captures the user-follower relationship of tweets propagation in Twitter.

Analysis of the Representation

Figure 4.2 depicts random tweet samples in hop-based cascade representation. The root node or the source tweet is located at the centre of the biggest cluster and all other nodes represents the successive retweets.

The following observations can be made,

- Maximum number of retweets are made directly from the source tweet.
- Most of the graphs have at least one dense cluster which does not include the source tweet i.e the claim has at least one very popular retweet.



(a) True claim

(b) True claim

(c) False claim

(d) False claim

Figure 4.2: Tweet samples in hop-based cascade representation

4.2.2 Time-based Structure

In this cascade representation, we calculate the time delay between a retweet and its source tweet. Using this delay, the retweet is positioned on the relevant stack using a sampling rate, refer Figure 4.3. The advantage of using this representation is the ease of using the temporal properties of the diffusion of a post. This representation effectively captures the life-cycle, popularity of a post and the amount of interactions accounted by the tweet over time. The sampling time (d) used in the structure is chosen to be 60 minutes for this study.

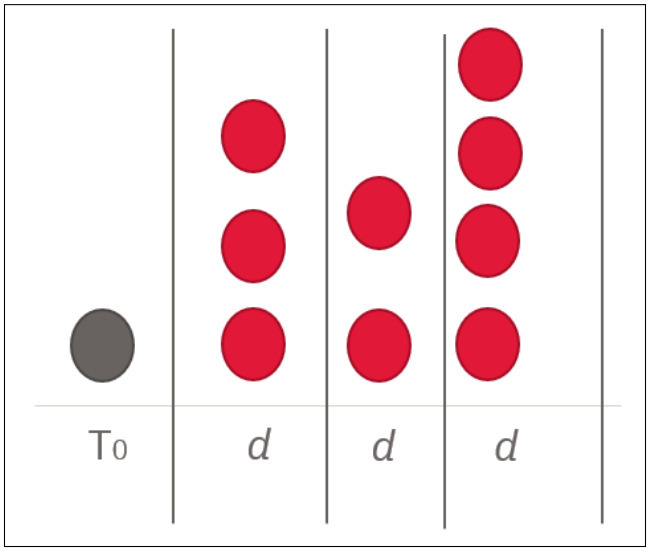
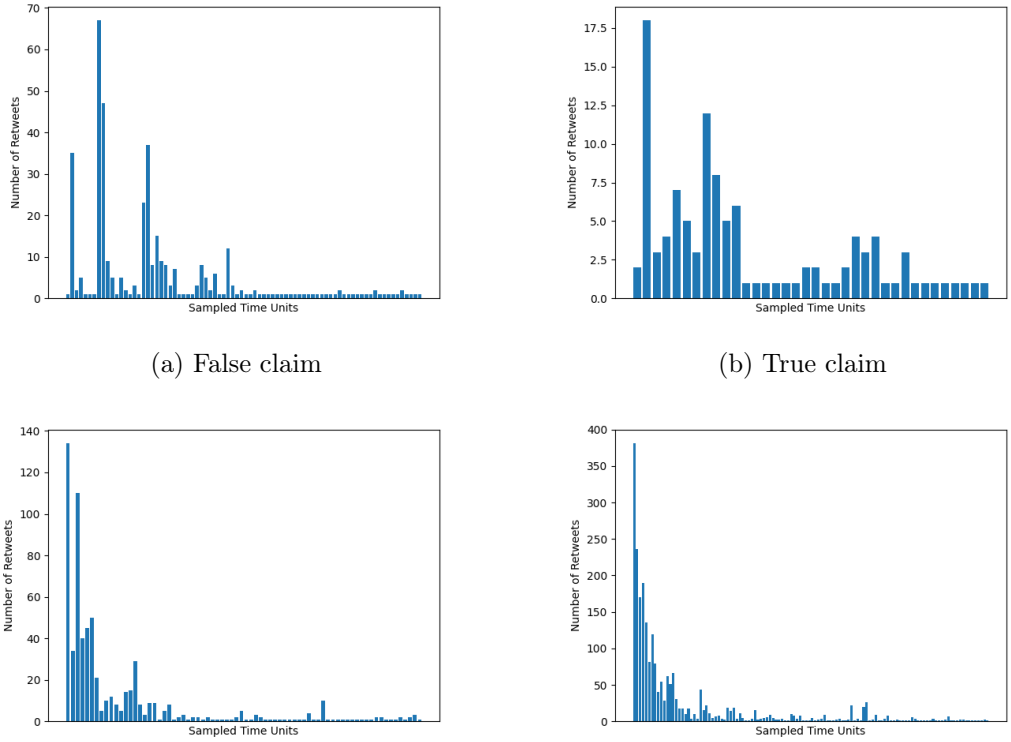


Figure 4.3: Time-based Representation

Analysis of the Representation

Figure 4.4 depicts random tweet samples in time-based propagation representation. The x-axis represents the time in units of sampling rate of 60 minutes and the y-axis represents the number of retweets or propagation flow.
The following observations can be made,

- During the first couple of hours the post had the farthest spread. That is the claim penetrated with more traction in the online social networks.
- Most of the plots follow a approximation of power law distribution.



(a) False claim

(b) True claim

(c) False claim

(d) True claim

Figure 4.4: Tweet samples in time-based cascade representation

4.3 Methodology

In this section, we provide a comprehensive overview of the methodology behind our proposed models. We first outline the steps involved in creating the propagation-based ensemble classifier, detailed in Section 4.3.1, which uses diffusion patterns to predict the veracity of tweets. Subsequently, in Section 4.3.2, we present its integration with the knowledge graph-based classifier introduced in Chapter 3, using fusion techniques.

4.3.1 Propagation-based Ensemble Classifier

In this study, along with using spatial features and temporal features, we got motivation from the field of epidemiology specifically, from the recent research related to COVID-19 and incorporated such features into our classification model.

The features used are detailed in Table 4.1 along with their respective feature types. These features can be easily obtained from the propagation tree of a post on Twitter. Furthermore, additional features can be added to our model if required.

Table 4.1: Feature Categorization

Number	Feature	Type
1	Number of Nodes	Spatial Feature
2	Total Diffusion Time	Temporal Feature
3	Total Peaks	Temporal Feature
4	Mean of timestamps delays	Temporal Feature
5	Basic Reproduction Number	Epidemiological Feature
6	Basic Transmission Rate	Epidemiological Feature
7	Super Spreaders	Epidemiological Feature
8	Growth Acceleration	Epidemiological Feature
9	Average Growth Speed	Epidemiological Feature
10	SD of Timestamps Delays	Temporal Feature
11	RMSSD of Timestamps Delays	Temporal Feature
12	Height	Spatial Feature

Number of Nodes

This represents the number of unique users that were involved in the diffusion of information on Twitter. Thus, for a claim dissemination pattern $C_i = \{R_i, reT_1, reT_2, .., reT_m\}$

$$\text{number of nodes} = \mathbf{card}|N_i| \tag{4.1}$$

where, R_i is the source tweet, reT_j is a retweet and $\mathbf{card}\{S\}$ is the number of elements of the set S.

Total Diffusion Time

This represents the total time taken for the information to propagate in the network, i.e. the life time of the claim on Twitter.

$$\text{Total Diffusion Time} = t(reT_m) - t(R_i) \tag{4.2}$$

Where, t(x) is the timestamp of the tweet object x and reT_m is the last retweet

Total Peaks

This feature pertains to the nodes that have a timestamp value exceeding the average timestamp across the entire diffusion cascade. Essentially, it captures nodes that are more recent or active compared to others. By tracking these "peak" nodes, we can gain insight into the recent behaviors or patterns within the data. Thus,

$$\text{TP} = \mathbf{card}\{v \in V \mid G(v, E)[\text{time}] > mean(G(V, E)[\text{time}])\} \tag{4.3}$$

Where, V are the nodes in the diffusion cascade G(V,E) and $\mathbf{card}\{S\}$ is the number of elements of the set S.

Mean of timestamps delays

Timestamp delay of a retweet is the difference between the source tweet timestamp and the retweet's timestamp, i.e., the amount of time the tweet was circulating before it was retweeted. We consider the the mean of the individual timestamps delays as a feature.

$$\overline{\text{timestamps delays}} = \frac{1}{V}\sum_{i=1}^{V} G(v,E)[\text{time}] - G(R_i,E)[\text{time}] \tag{4.4}$$

Where, V is the total number of nodes. R_i is the source tweet of the claim and G(V,E) is diffusion cascade.

Basic Reproduction Number

In epidemiology, the basic reproduction number is the expected number of cases directly generated by one case in a population where all individuals are susceptible to the disease [106]

More precisely, it is the number of direct infections produced by an infected individual in a population that is totally susceptible. This number is important in determining how quickly a disease will spread through a population. An infectious disease can be considered analogous to the spread of a claim in a social network. For this study we define the basic reproduction number (R0) as the number of retweets directly from the source tweet (R_i),

$$\text{R0} = \mathbf{card}\{e \in E \mid e \in G(V,e) \wedge G(R_i,e)\} \tag{4.5}$$

Where, R_i is the source tweet and G(V,E) is the diffusion cascade. Finally, $\mathbf{card}\{S\}$ is the number of elements of the set S.

Basic Transmission Rate

The Susceptible-Infectious-Recovered (SIR) model is a simple model for the spread of a infectious disease. The basic transmission rate (denoted β) is defined as the number of effective contacts made by an infected person per unit time in a given population. In this study, we interpret basic transmission rate as the number of retweets made during the first day (Td) of the source tweet.

$$\beta = \mathbf{card}\{v \in V \mid G(v,E)[\text{time}] \leq G(R_i,E)[\text{time}] + Td\} \tag{4.6}$$

Where, Td is 24 hours, R_i is the source tweet of the claim and G(V,E) is diffusion cascade. Finally, $\mathbf{card}\{S\}$ is the number of elements of the set S.

Super Spreaders

In an investigation conducted to analyze the transmission of coronavirus infections [107], researchers observed a significant impact on the spread of the virus were attributable to

individuals identified as 'super spreaders'. Super spreaders are individuals with greater than average propensity to infect. Within this study, we delineate super spreaders (SS) as the number nodes exhibiting an edge count exceeding the average number of edges.

$$\text{SS} = \mathbf{card}\{v \in V \mid G(v, E) > mean(E)\} \tag{4.7}$$

Where, G(V,E) is the diffusion cascade and $\mathbf{card}\{S\}$ is the number of elements of the set S.

Growth Acceleration

In epidemiology, growth acceleration is defined as the $(cases \setminus day^2)$. Recently, in a study it has been shown that growth speed and growth acceleration are very effective for the analysis of the COVID-19 pandemic [108].

In this study, we consider the edges i.e the retweets as the cases and define growth acceleration (GA) as follows,

$$GA = \sum_{i=1}^{V} \frac{1}{(G(v, E)[\text{time}] - G(R_i, E)[\text{time}])^2} \tag{4.8}$$

Where, R_i is the source tweet of the claim and G(V,E) is diffusion cascade and V is the total number of nodes.

Average Growth Speed

Additionally, as mentioned previously Growth speed was also shown to be very effective in the analysis of the COVID-19 pandemic [108]. In this study, we define average growth speed (avgGS) as follows,

$$avgGS = \frac{heightG(V, E)}{\overline{\text{timestamps delays}}} \tag{4.9}$$

Where, $\overline{\text{timestamps delays}}$ is the average of all the timestamp delays and G(V,E) is diffusion cascade.

Standard Deviation of Timestamps Delays

We also consider the standard deviation of the timestamps delays of the child nodes. Using this feature we take into account the measure of the spread of values from the mean.

$$\sigma = \sqrt{\frac{1}{V-1}\sum_{i=1}^{V}(t_i - \bar{t})^2} \tag{4.10}$$

Where, t are the individual timestamp delays of each retweet from the source tweet and V is the total number of nodes.

RMSSD of Timestamps Delays

We also consider the root mean square of successive differences between retweet timestamps (RMSSD). In medical science, RMSSD is considered the primary time domain measure used to estimate the vagally mediated changes [109]. RMSSD reflects the peak-to-peak variance in a time series data. As mentioned in Section 4.2.2, during the initial stages of propagation, claims exhibit the widest spread, with minimal successive differences between retweets. Consequently, we integrated RMSSD as a feature in our model to capture early-hour changes in propagation flow.

$$\text{RMSSD} = \sqrt{\text{mean}\{\text{diff}\{t1, t2, .., tN\}^2\}} \tag{4.11}$$

Where, t are the individual timestamp delays of each retweet from the source tweet.

Height

This represents Height of the diffusion tree i.e. the length of the path from source tweet to its farthest retweet node in the hop-based representation of the diffusion pattern.

$$\text{Height} = \sum_{R_i}^{reT_n} 1 \tag{4.12}$$

where, R_i is the source tweet and reT_n is the farthest retweet.

Propagation-based Classification Model

For this study we used a voting classifier, which is an ensemble machine learning classifier that trains various base models or estimators and predicts on the basis of aggregating the findings of each base estimator. Voting classifiers has been shown to reduce the aggregate errors of a variety of the base models and increase final accuracy [110]. The aggregating criteria used in this study is hard voting which is the combined decision of the class label that has been predicted most frequently by the classification models.

The base models used are as follows, as shown in Figure 4.5:

- KNN classifier
- Decision tree classifier
- Multi-layer perceptron classifier

Where, the predicted class label $\hat{y}$ of our classifier is as follows,

$$\hat{y} = \text{mode}\{C_1(x), C_2(x), C_3(x)\} \tag{4.13}$$

Where, $C_j(x)$ is the predicted class label of classifier j.

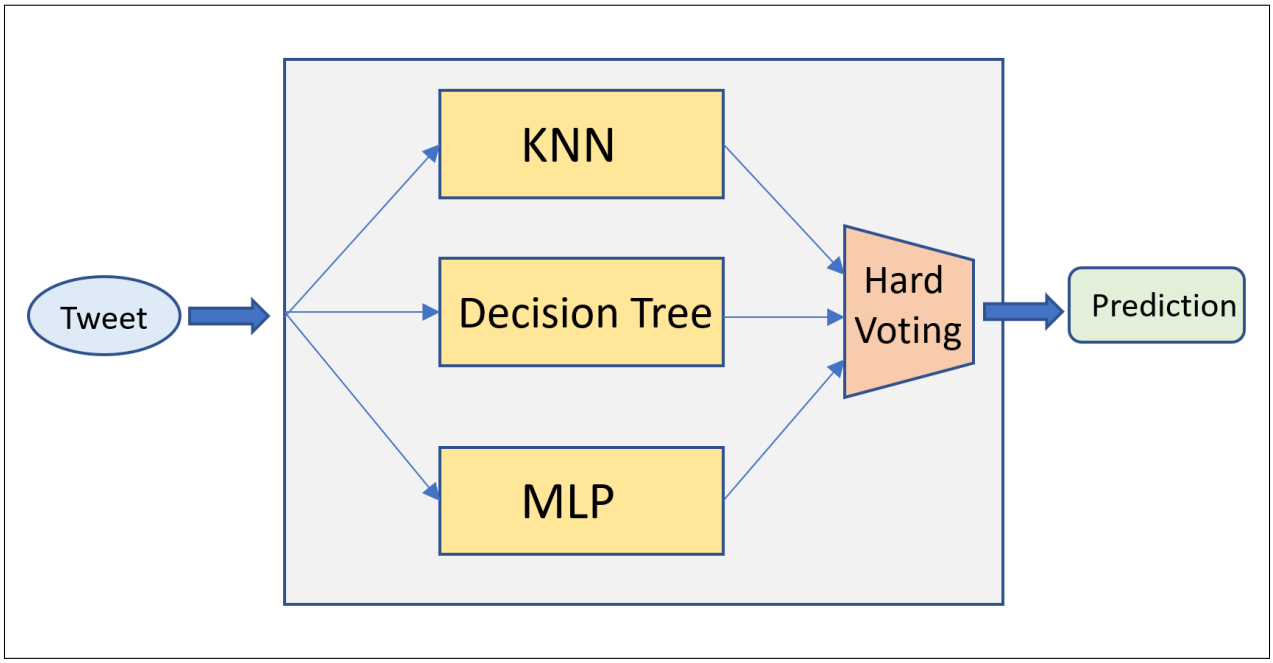


Figure 4.5: Voting classifier diagram

4.3.2 Multi-modal Fusion with KG-FC Model

In this section, we detail the integration of the propagation-based ensemble classifier, as outlined in the previous section, with the knowledge graph-based classifier introduced in Chapter 3. This integration uses multi-modal fusion techniques. Studies in multi-modal fusion have established that models trained by merging data from multiple sources outperform those trained with just one source [111], [105], [112]. The idea is that individual data sources can provide different kinds of information which resolves ambiguity, improve the overall quality of noisy data, or enable the exploitation of correlations [105].

One of the challenges encountered in the field of multi-modal machine learning [113] is devising techniques for integrating diverse modalities. These various approaches can be broadly classified into two categories: early fusion and late fusion [105], depending on where the fusion occurs within the processing pipeline. Additionally, hybrid fusion methods strive to amalgamate the attributes of these two fundamental approaches [105].

In our research, we investigate two fusion methodologies aimed at integrating the content (knowledge graph-based) and propagation-based models: early fusion and late fusion. Early fusion entails the integration of data from multiple modalities at the initial stages of model development, facilitating a comprehensive understanding of the input data right from the outset [105]. Studies argue that early fusion captures more nuanced interactions between modalities and enhances the quality of acquired representations [111]. In contrast, late fusion combines the outcomes of distinct models, allowing them to specialize in their respective

modalities before consolidating their insights at a later stage. It combines data following separate full-scale processing in distinct unimodal streams [105].

Early Fusion

In the early fusion-based approach depicted in Figure 4.6, vector representations from both the knowledge graph-based (content) and the propagation-based (diffusion pattern) models are concatenated. These concatenated vectors subsequently serve as input to the SVM classifier. The knowledge graph model contributes four dimensions, while the propagation model provides three dimensions, one from each of its base models. This distribution of the dimensions ensures that neither modality dominates the other.

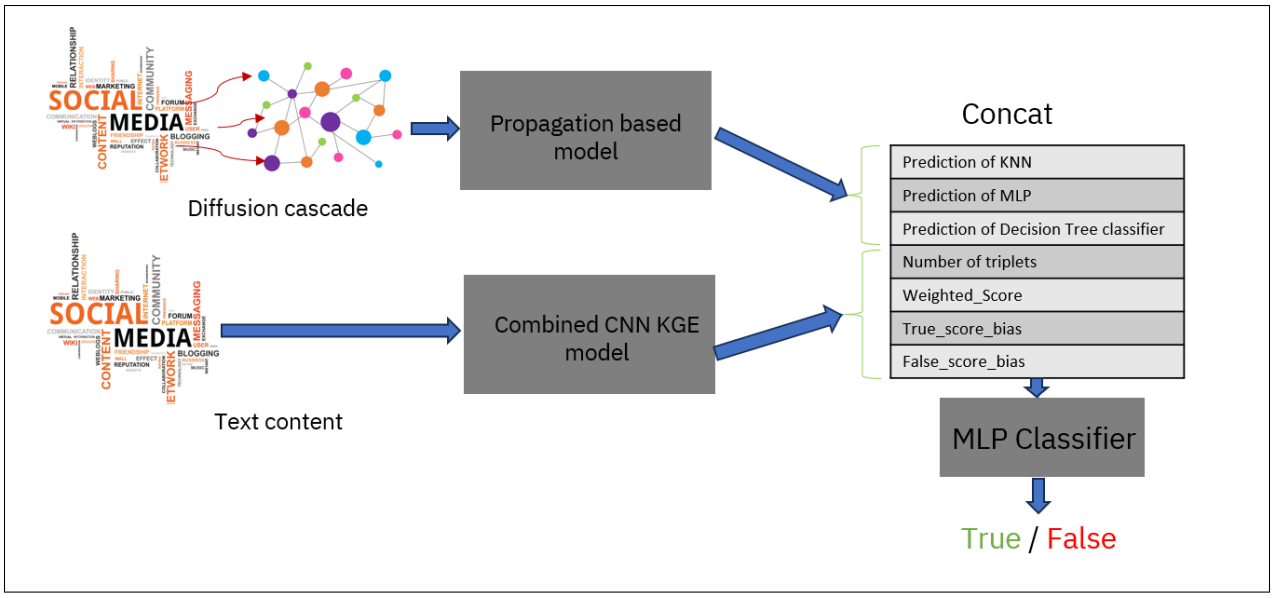


Figure 4.6: Early fusion strategy

Late Fusion

Contrastingly, the late fusion method derives its final prediction by aggregating the outcomes from the base classifiers. We explore the mean aggregation strategies for late fusion using the class confidence scores. Figure 4.7 further delineates the architecture of our late fusion approach.

The class confidence score of the knowledge graph-based classifier (C_{KGC}) is,

$$C_{KGC} = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} tcs_i & \text{if } tcs_i > fcs_i \\ 1 - fcs_i & \text{if } fcs_i > tcs_i \end{cases} \tag{4.14}$$

Where, (tcs_i, fcs_i) are the combined score for triplet t_i and n is the total number of triplets in the claim.

The class confidence score for the propagation-based classifier is as follows,

$$c_{PbC} = \frac{1}{3}\sum_{i=1}^{3} \begin{cases} 1(C_i(x) = \hat{y} = \text{true}) \\ 1(C_i(x) \neq \hat{y} = \text{false}) \end{cases} \tag{4.15}$$

Where, $\hat{y}$ predicted class label of the ensemble propagation-based classifier and $C_i(x)$ is the predicted class label of the i-th classifier.

Finally, the prediction for the late fusion mean aggregator is as follows,

$$c_{\text{fusion}} = \left\lfloor \frac{c_{PbC} + c_{KGC}}{2} \right\rfloor \tag{4.16}$$

Where, $\lfloor . \rfloor$ is the floor function and c_{PbC} is the class confidence score of the propagation-based classifier, refer Equation 4.15. c_{KGC} is the class confidence score of the knowledge graph-based classifier, refer Equation 4.14.

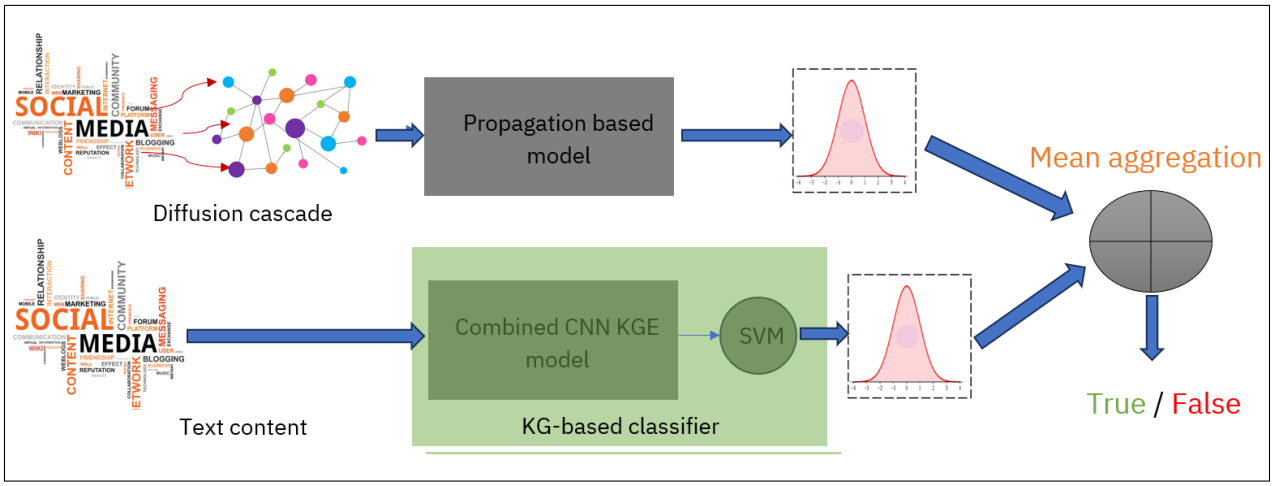


Figure 4.7: Late fusion strategy

4.4 Datasets

For evaluation of our model, we use two popular publicly available datasets [63], Twitter15 and Twitter16, which have been widely adopted as standard data in the field of misinformation detection. The dataset contains source claims along with the source tweet and corresponsind retweets with timestamps. Each claim is annotated with one of the four class labels, i.e., 'unverified rumor', 'false rumor', 'true rumor' and 'non-rumor'. Some important characteristics of the dataset are mentioned in Table 4.2.

Table 4.2: Basic Statistics of the Datasets

Statistics	**Twitter15**	**Twitter16**
Number of Users	306,402	168,659
Number of Tweets	331,612	204,820
Max. Number of retweets	2,990	999
Min. Number of retweets	97	100
Avg. Number of retweets	493	479

As our study focuses on binary classification, we re-annotated to two class labels. We considered the 'true rumours' as true class label and 'false rumours' as false claims. Additionally, we disregarded the data labeled as 'unverified rumour' and 'non-rumours' as they introduced ambiguity and could potentially distort the results. There are a total of 575 fake samples and 579 true samples in the datasets, resulting in an almost balanced 50:50 class distribution. Finally, For a fair comparison,we randomly choose 80% data for training and 20% for testing.

4.4.1 Data Preparation

We normalized the features by scaling and translating each feature individually such that it is in the given range on the training set, between zero and one. We used the min-max normalization method as depicted in the equation below,

$$y = \frac{x - min}{max - min} \tag{4.17}$$

Where, x is the initial value of the considered feature, y is the final normalized value of the feature, min is the minimum value of the feature and max is the maximum value of the feature.

4.5 Explainability of the Model

This section describes the explainability of the propagation-based model and the fusion (late) model. By enhancing transparency, we aim to foster trust and facilitate a deeper understanding of the model's behavior in various scenarios.

4.5.1 Propagation-based Ensemble Classifier

Studies have shown that intrinsic XAIs provide better explanations than post-hoc XAIs [79], however has a trade-off with accuracy. The purpose of this study is to provide intrinsic model explanations while keeping the accuracy high. the propagation-based method balances intrinsic model explanations via KNN and decision tree (DT) classifiers while maintaining high accuracy.

The following two methods were used to accomplish this,

- Firstly, the unique design of the the propagation-based ensemble classifier asserts that at least one intrinsic explainable classifier is part of the final aggregated vote. That is, in the best-case scenario both the intrinsic explainable classifiers (KNN or DT) have the same predicted label. Whereas, in the worst-case scenario along with MLP classifier either KNN or DT classifier has the same predicted label which can be used to provide intrinsic explainability.
- Secondly, we added MLP to balance the accuracy-explainability trade-off in intrinsic XAIs. Though, MLP is not an intrinsic explainable algorithm on its own, the combination the three classifiers provides intrinsic explainability along with high accuracy.

KNN Classifier

KNN is a transparent machine learning technique. In terms of system interpretability, KNN estimated values are based on the notion of similarity between instances and distance. KNN is easily interpretable by nature because the nearest neighbors themselves have explanations that are readily understood by humans as they lie in the input domain.

For providing human readable explanations for a particular predicted class label, we employed the following steps:

1. Collect nearest K neighbours of considered point (P).
2. Filter out same class neighbors of P, which are inherently higher in number.
3. Project a new point (P_{new}) by calculating the arithmetic mean filtered points.
4. Get the four highest correlated features between P_{new} and P using Manhattan distance.
5. Display the number of nearby same class label neighbours and the highest correlated features.

Decision Tree Classifier

A decision tree provides a hierarchy of very specific questions and predicts outcomes based on decision rules (if-then-else rules). The answer to one question guides the prediction process down various branches of the tree. At the bottom of the tree is the prediction. Hence, for interpretations, we review decisions by traversing top-to-bottom tree paths and noting question responses for explanations.

To this direction we used the following steps,

1. Fetch the decision rules from the classifier.
2. Use the rules to showcase the answers the specific rule addresses.
3. Every rule corresponds to one feature, delivering a local explanation for that feature's value.

4. For a given point (P), traversing the decision tree from top to bottom reveals explanations for the predicted class label. Inherently the number of the explanations is the depth of the decision tree.

Sample Explanation from Propagation-based Classifier

Figure 4.8 displays the explanations generated by the propagation-based ensemble model on a random data point (P), where KNN classifier and DT classifier had the same predicted class label. We can observe that, three explanations were generated for the decision tree classifier, which is logical as the depth of the tree was three. Furthermore, for point P, the KNN classifier interpretations were made from the seven nearby fake tweets out of the ten neighbours. An interesting observation can also be made that explanations for both the classifiers almost correspond for the same statistical properties of the propagation cascade.

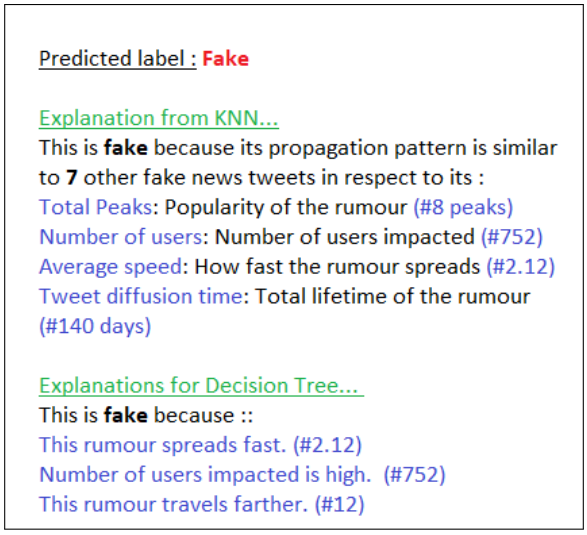


Figure 4.8: Explanation generated for a random sample from the propagation-based classifier

4.5.2 Explainability of Fusion Classifier

The explainability of the fusion (late), combines the interpretations from the propagation-based model and the knowledge graph-based model, as discussed in Chapter 3. Figure 4.8 provides a sample explanation generated from the late fusion classifier for the same point P.

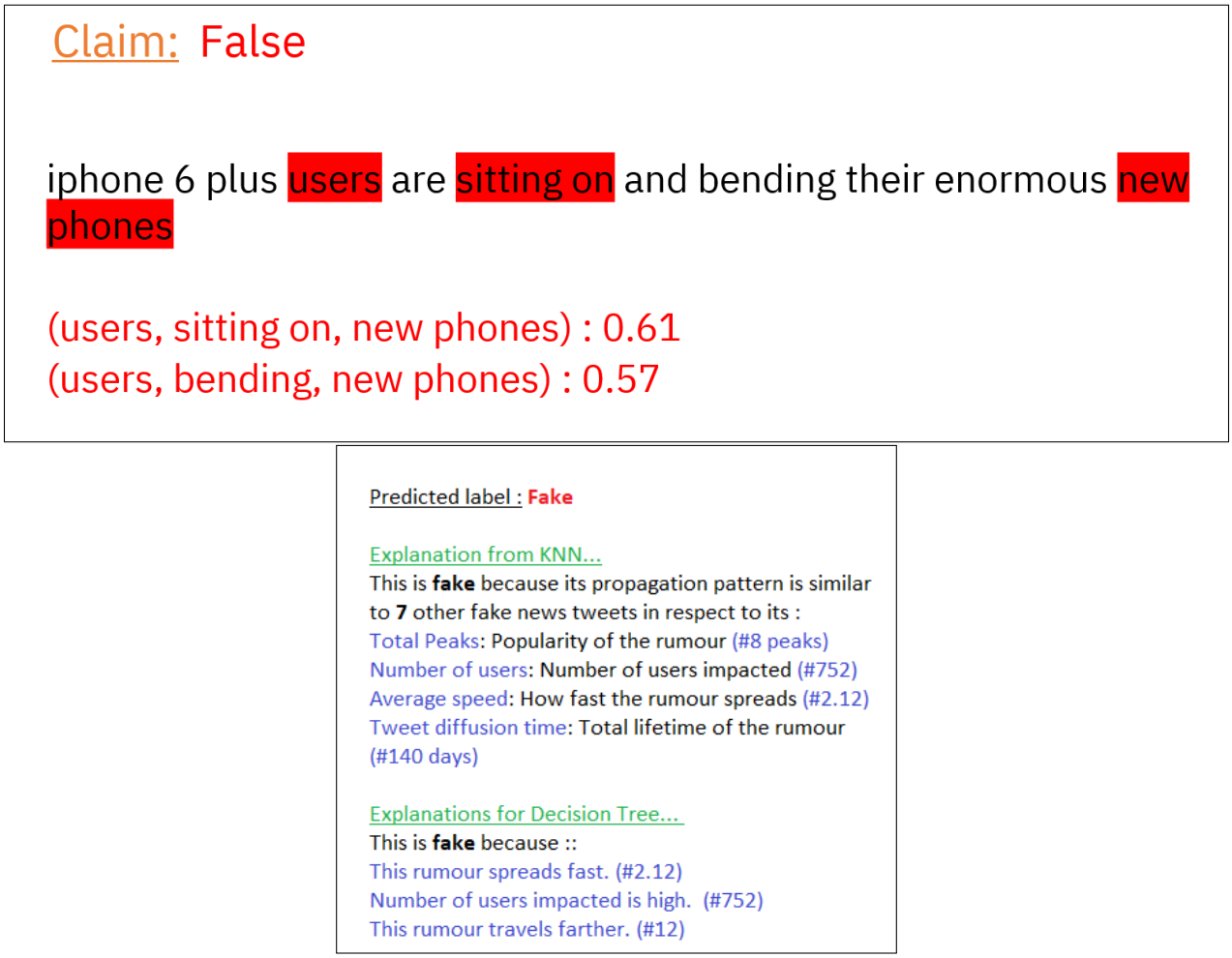


Figure 4.9: Explanation generated for a random sample from the late fusion classifier

4.6 Evaluation

In this section we discuss the results of the propagation-based classifier and present the evaluation of the fusion classifiers. Furthermore, we compare the models with five state-of-the-art baselines misinformation classification models and provide the results of the ablation study.

4.6.1 Metrics

Let TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively. We employed four evaluation criteria extensively used in text classification tasks,

namely accuracy, precision, recall and F1 score (calculated as in the equations below), to evaluate the prediction performance of the models:

- Accuracy (A): Accuracy is a measure of the classifier's ability to correctly classify a piece of claim as either true or false.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.18}$$

- Precision (P): Precision is a measure for the classifier's exactness such that a low value indicates a large number of false positives. The precision represents the number of positive predictions divided by the total number of positive class values predicted.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.19}$$

- Recall (R): Recall is considered a measure of a classifier's completeness (e.g., a low value of recall indicates many false negatives), where the number of true positives is divided by the number of true positives and the number of false negatives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.20}$$

- F1 score (F1): The F1 score is calculated as the weighted harmonic mean of the precision and recall measures of the classifier.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.21}$$

4.6.2 Results and Discussion

In this section, we first illustrate the experimental setup and the baselines. Then, we show the observed results and comparison with various baseline models. All experiments are conducted on a machine with Intel(R) Core(TM) i7-8750H 2.20GHz CPU, 16 GB of RAM and NVIDIA GeForce RTX 4090 GPU running Windows 11. For the knowledge graph-based classifier, the ConvE models used the same hyper-parameters as discussed in Section 3.5.3 of Chapter 3.

4.6.3 Propagation-based classifiers

KNN Classifier

The number of neighbors used in this model is ten. Table 4.3 shows the results from the k-nearest neighbors classifier with varying hyper-parameters associated with the model. It can be discerned from the table the model with Manhattan as the distance metric performs the best, with the highest accuracy and recall and a good overall precision. This is rational

as studies have shown that Manhattan distance (L1 norm) ususally performs better than common distance measures in the case of high dimensional data [114].

Table 4.3: Results of the KNN classifier on Twitter16

Distance metric	Accuracy	Precision	Recall	F1 Score
Cosine	0.812	0.844	0.879	0.861
Manhattan	**0.813**	**0.859**	0.887	**0.873**
Correlation	0.805	0.789	**0.893**	0.838
Euclidean	0.810	0.809	0.879	0.843
BrayCurtis	0.790	0.783	0.881	0.829

Decision Tree Classifier

The maximum depth of the tree is chosen to be three, in order to reduce computational complexity. Table 4.4 depicts the results with varying hyper-parameters. It can be observed that the results are better with the entropy splitting criterion. However, the accuracy is almost the same for both the splitting methods. This is reasonable as the internal working of both the splitting methods are very similar. Nevertheless, for the ensemble model we choose the entropy criterion.

Table 4.4: Results of the Decision Tree classifier on Twitter16

Splitting criterion	Accuracy	Precision	Recall	F1 Score
Gini	0.811	0.855	0.868	0.861
Entropy	**0.812**	**0.868**	**0.882**	**0.875**

Multi-layer Perceptron Classifier

The MLP classifier has been configured with two hidden layers containing (5,2) units respectively. Limited-memory BFGS algorithm has been used for weight optimization as it converges faster and performs better for small datasets. Table 4.5 depicts the results. It can be observed from the results that the accuracy and recall is highest with the sigmoid activation function. Where, sigmoid function is defined as,

$$f(x) = 1/(1 + exp(-x)) \tag{4.22}$$

As the number of hidden layers is very low in this model the vanishing gradient problem does not play a significant role and hence the accuracy using sigmoid function is higher compared to other activation functions.

Table 4.5: Results of the MLP classifier on Twitter16

Activation function	Accuracy	Precision	Recall	F1 Score
Tanh	0.795	0.771	**0.912**	0.836
Sigmoid	**0.823**	**0.857**	0.891	**0.874**
ReLU	0.812	0.842	0.862	0.852

Propagation-based Ensemble Classifier

The results for the individual classifiers are shown in table 4.6 with the optimum parameter configurations. Firstly, It can be observed that MLP classifier has the highest accuracy of 82.31%, however the precision is low with 0.8574.

Table 4.6: Results of the individual classifiers on Twitter16

Type	Accuracy	Precision	Recall	F1 Score
KNN classifier	0.813	0.859	0.886	0.873
MLP classifier	**0.823**	0.857	**0.891**	0.874
Decision tree classifier	0.812	**0.868**	0.882	**0.875**

Secondly, the KNN classifier also has a high accuracy of 81.29% with the low recall and precision.

Finally, the precision is highest in the case of the decision tree classifier with 0.8679, with an accuracy almost similar to that of the KNN classifier. Thus, it can be reasoned that a combination of these three classifiers might produce better results. This motivated us to implement an ensemble voting classifier for this study.
Table 4.7 depicts the results for the ensemble voting classifier.

Table 4.7: Results of the Ensemble Voting Classifier on Twitter16

Type	Accuracy	Precision	Recall	F1 Score
Propagation-based classifier (PbC)	**0.852**	**0.884**	**0.892**	**0.888**

It can be inferred from Table 4.7 that the accuracy has increased to 85.22 % with the use of the voting classifier. Ensemble methods like the voting classifier are ideal for reducing the variance in models, thereby increasing the accuracy of predictions. The variance is eliminated when multiple classifiers are combined to form a single prediction. Additionally, it can be observed that the precision and recall of the voting classifier are also highest with 0.8843 and 0.8917 respectively.

From our experiments, it can be reasoned that the propagation-based ensemble classifier outperformed all individual models.

4.6.4 Ablation Study

To study the contribution of each feature type to the propagation-based classifier (PbC), we carry out the ablation experiments in this part. The experimental results are shown in Table 4.8. The ablation experiments include the following three variants of our classifier:

- without spatial features: removing the spatial features of the propagation-based classifier.
- without temporal features: removing the temporal components of the propagation-based classifier.
- without epidemiological features: removing the epidemiological based features of the propagation-based classifier.

Table 4.8: Results of the ablation experiments using Twitter16

Type	Accuracy	Precision	Recall	F1 Score
w/o Spatial	0.821	0.823	0.808	0.815
w/o Temporal	0.826	0.859	0.881	0.870
w/o Epidemiological	0.771	0.780	0.828	0.803
PbC	**0.852**	**0.884**	**0.892**	**0.888**

From Table 4.8, we can observe that all ablation variants drop some accuracy compared with the primary model. Specifically, when removing the spatial features, the accuracy drops by 3.1%, the precision and recall also dropped. The replacement of the temporal features caused the accuracy to decrease by 2.7% with lower precision and recall. However, the accuracy drop was most significant when the epidemiological features were removed, accounting to 8.1% along with lowest precision and recall. This corroborates that epidemiological features inspired from the study on COVID-19, play an essential role for misinformation detection using propagation cascades. In conclusion, overall the primary model, with the three component types involved, provides a better choice compared to the ablation variants.

4.6.5 Baseline Model Comparison

We compared our proposed model with the following five state-of-the-art baselines misinformation classification models,

1. tCNN [115]: a modified convolution neural network that learns the local variations of user profile sequence, combining with the source tweet features.
2. CRNN [116]: a state-of-the-art joint CNN and RNN model that learns local and global variations of retweet user profiles, together with the resource tweet.

3. CSI [117]: a state-of-the-art misinformation detection model incorporating articles, and the group behavior of users who propagate misinformation by using LSTM and calculating the user scores.

4. dEFEND [118]: a state-of-the-art co-attention-based misinformation detection model that learns the correlation between the source article's sentences and user profiles.

5. GCAN [119]: a state-of-the-art graph-aware co-attention network based misinformation classifier that uses user profiles metadata, news content and propagation pattern.

Table 4.9 compares our approachs to the industry standards. It can be inferred that the proposed propagation-based classifier(PbC) outperforms most of the state-of-the-art approaches on both datasets in terms of accuracy while attaining highest precision and recall. In particular, PbC achieves an accuracy of 84.47% and 85.22% on the datasets respectively. Although GCAN achieved the highest accuracy over PbC, the precision and recall are very low. Whereas, PbC received at par accuracy with GCAN with higher precision and recall. Furthermore, GCAN in addition to propagation pattern uses the user profile metadata and tweet content. Whereas, PbC solely uses the diffusion pattern to create a classifier.

While both fusion methods benefited from the integration of the knowledge graph-based model, their prediction performance surpassed that of all other models. Specifically, the late fusion method yielded accuracies of 92.37% for the Twitter15 dataset and 91.12% for the Twitter16 dataset, marking improvements of 7.90% and 5.90%, respectively, over the standalone PbC model. This underscores the value of incorporating the knowledge graph-based model. Although the early and late fusion classifiers showed similar accuracy, we favored the late fusion approach due to its simplicity. Additionally, as highlighted in [120] late fusion classifiers have enhanced robustness to noise and outliers compared to early fusion classifiers.

Table 4.9: Experimental results on Twitter15 (T15) and Twitter16 (T16) datasets

Method	**Recall**		**Precision**		**Accuracy**		**F1 Score**	
	T15	T16	T15	T16	T15	T16	T15	T16
tCNN [115]	0.520	0.626	0.519	0.624	0.588	0.737	0.520	0.625
CRNN [116]	0.530	0.643	0.529	0.641	0.591	0.757	0.530	0.642
CSI [117]	0.686	0.630	0.699	0.632	0.698	0.661	0.692	0.631
dEFEND [118]	0.661	0.638	0.658	0.636	0.738	0.701	0.659	0.637
GCAN [119]	0.829	0.763	0.825	0.759	0.876	0.908	0.827	0.761
PbC	0.851	0.892	0.857	0.884	0.845	0.852	0.854	0.888
KG-FC	0.872	0.911	0.866	0.892	0.849	0.870	0.869	0.902
PbC + KG-FC (early fusion)	0.914	0.918	**0.925**	0.897	0.918	0.909	**0.920**	0.908
PbC + KG-FC (late fusion)	**0.915**	**0.942**	0.911	**0.899**	**0.924**	**0.911**	0.913	**0.920**

4.7 Limitations of the Model

The following are the limitations of our model,

- While our approach shows promise, its applicability to other online social networks and languages may vary. Generalizing across different contexts my be difficult. To address this problem, we intend to extend the work to other online social networks in future studies.
- Ensembling can sometimes lead to overfitting, particularly when combining multiple models. To address this, we chose diverse base models with varied architectures, hyperparameters, and features. This diversity lowers the risk of overfitting, as different models are less likely to make the same training data errors. We also used hard voting criteria to combine model predictions for the propagation-based classifier, reducing the risk of overfitting to any single model's output [121].
- Our model's explanations are designed to assist human users, but the ultimate judgment on the veracity of claims still rests with human users, who may have their own biases. Future studies will use quantitative and qualitative measures to evaluate the generated explanations.

4.8 Chapter Summary

Our study illustrates how the propagation-based ensemble classifier uses the unique diffusion structure of tweets to increase the accuracy of misinformation detection on Twitter. With insights from epidemiological models and recent COVID-19 research, we integrated diverse features into the propagation-based classifier, enabling it to outperform existing propagation-based models. The model design was further augmented with fusion models, melding the strengths of both early and late fusion classifiers. After evaluating both techniques, we find that the late fusion method significantly outperforms the baseline models. Furthermore, we provide explanations to enable users to understand the rationales behind predicted class labels.

This study can be extended in the following direction:

- Incorporating both quantitative and qualitative measures into the evaluation of generated explanations, such as A/B testing, user surveys, and expert evaluations.
- Investigating methods to understand the echo chamber effect, where beliefs are amplified within a closed system, particularly in the diffusion cascades of a post and their use in misinformation detection.
- Expanding the scope of the study to include other online social networks, such as Facebook and Instagram.

Chapter 5

Fact Checking on Reddit Using Knowledge Graphs

In this chapter, we describe the proposed transformer-based model that converts SNSW to standard English and the performance evaluations. This translator model is combined with the KG-based fact checking model presented in Chapter 3 to improve misinformation identification on Reddit, where posts often contain a lot of SNSW.

The chapter is structured as follows: Section 5.1 details in importance of misinformation identification on Reddit and the motivation of our work. Section 5.2 discusses the background on SNSW. Section 5.3 details the methodology used by our models. In Section 5.4 we discuss the datasets used for the study. Section 5.5 presents the evaluation our model using different metrics. Section 5.6 lists the limitations of the models. Finally, we summarize the chapter and discuss future directions of the study in Section 5.7.

5.1 Introduction

In an age of digital revolution and interconnectivity, the exponential growth of online platforms and user-generated content has transformed the way information is disseminated and consumed globally. This shift is particularly evident in the rise and dominance of online social networks (OSNs). OSNs have become primary conduits for news, updates, and diverse viewpoints, reshaping our consumption patterns and influencing public opinion. Among various OSNs, Reddit[4] and Twitter[5] are paramount, each distinguished by its unique content delivery and user engagement paradigms [122]. Reddit, often referred to as "The Front Page of the Internet" [123], is a space where users can submit content, either as text posts or direct links, and engage in discussions surrounding that content. It is structured into areas of interest known as "subreddits", each honed into specific topics and governed by its own set of rules. Reddit nurtures deep, community-centric interactions. On the other hand, Twitter

[4]www.reddit.com

[5]www.twitter.com

adopts a micro-blogging format, underscored by a user-follower relationship. Users on Twitter broadcast concise, real-time messages to their followers, placing emphasis on instantaneous information sharing and interaction.

Reddit and Twitter, though both influential platforms in the realm of digital communication, present different challenges and dynamics when it comes to the dissemination of claims. While Reddit boasts a blog-like, topic-centric structure, Twitter leans towards a user-centric model. These core differences play a crucial role on how claims flows and the inherent challenges tied to fact checking within each environment. Misinformation detection on OSNs is pivotal, not just for the accuracy of data but also for the societal implications it carries [124]. False narratives can skew public perceptions, influence policy decisions, and even alter the course of events in real-time [125, 124].

According to Semrush.com, which provides real-time analytics and ranks websites based on search traffic, Reddit was the third most visited site in the U.S [126] and the eight most popular worldwide [127] in July 2023. This prominence underscores the essential task of addressing misinformation on the platform. Reddit's design promotes in-depth discussions. While this fosters comprehensive viewpoints, it also risks magnifying inaccuracies or falsehoods. In our rapidly evolving digital landscape, the need to distinguish factual content from misinformation on such high-traffic platforms is paramount. This brings us to the topic of fact checking on Reddit.

Typically, content-based fact checking methods take center stage for misinformation detection [15]. Among these methods, one approach that has gained significant attention is the use of knowledge graphs [8], [39]. Knowledge graphs are structured representations of facts that capture relationships between entities and their attributes. In this chapter, we use the knowledge graph-based classifier, introduced in Chapter 3, to facilitate fact checking on Reddit.

However, conversational language in platforms like Reddit, which resemble blogs, frequently integrates jargon, slangs and colloquial terms specific to Reddit [51, 52]. One of the challenges in NLP is the ability to understand and process out-of-vocabulary(OOV) words [128]. Conversational language used in blogs often contains such slang and non-standard words (SNSW). These unconventional terms are not typically present in curated datasets used to train NLP models. As a consequence, it becomes difficult for traditional models to accurately interpret and analyze text data with SNSW [53, 54].

For instance, with our knowledge graph-based classifier, the accurate extraction of triplets from an article can be hampered if SNSW is present. Triplets, which consist of entities and their relations, form the core of our classifier's methodology for predictions. However, SNSW terms in articles can often refer to specific entities and relations, making them crucial to the triplet formation. Not properly processing the text due to SNSW means missing out triplets. Therefore, translating these SNSW terms into standard English is essential for ensuring a thorough extraction of all relevant triplets.

To address this challenge, it is essential to first define and identify SNSW. In this study, we define SNSW as words or phrases that are unique to urban areas, including slang, idioms, jargon, abbreviations, and other specialized terminology that may not be widely recognized

or understood. Examples of SNSW may include terms such as "hustle," "lit," "fam," "TBH" (to be honest), or "ICYMI" (in case you missed it), which can have different meanings or connotations within the urban context than in other settings.

In this chapter, we use the knowledge graph-based classifier, to facilitate fact checking on Reddit. Recognizing the challenges posed by SNSW, we introduce a dedicated translator that converts text with SNSW into standard English. This translated content not only serves as an input for our classifier but can also be used by other downstream NLP models or human reviewers, thereby enhancing comprehension and readability. We discuss the methodologies employed, the inherent challenges, and the broader implications of our models. Through this study, we strive to enhance the veracity of digital platforms, ensuring users are equipped with trustworthy claims in the vast expanse of the digital universe.

5.2 Background on SNSW

An inherent challenge in processing conversational language, especially on platforms like Reddit, is handling slang and non-standard words (SNSW). These unconventional terms often deviate from the vocabulary found in standard linguistic resources, making them problematic for conventional natural language processing (NLP) models.

Early attempts to solve this problem relied on handcrafted rules or dictionaries that were manually curated by linguists or domain experts [129]. However, these methods had limited effectiveness as they often failed to capture the nuances and variations of language, and had a limited number of entries. With the advent of machine learning techniques, researchers have been exploring data driven methods for analysis of SNSW. These methods encompass a range of approaches, including the use of traditional classifiers like SVM [130], as well as the application of neural networks [131].

However, the data driven methods have limitations. Slang terms and phrases can evolve rapidly and may have highly context dependent meanings, making it difficult to collect and label enough data for effective model training [130]. Additionally, these models may be vulnerable to overfitting, as the distribution of slang words in the training data may not reflect their real-world distribution [131]. Furthermore, models that use encoders can be less flexible and inaccurate even when dealing with common words [132].

To address the above gaps we introduce SNSW-DB, an automatically curated, regularly updated, large-scale lexical database from crowdsourced dictionaries. The database contains SNSW terms, example sentences, and their standard English synonyms generated by our synonym generator module. We also propose a transformer-based translation model that converts articles with SNSW to standard English. Translated articles can be used by human reviewers to enhance readability and utility, or by downstream NLP models to improve their classification performance.

SNSW-DB is regularly updated using online crowdsourced dictionaries, Urban Dictionary[6]

[6]https://www.urbandictionary.com/

and Wiktionary[7]. By using up-to-date, large-scale, and crowdsourced dictionaries, we provide a comprehensive and continuously updated database, overcoming the limitations of handcrafted dictionaries and data driven approaches.

The translation model serves as an abstract interface that bridges the gap between NLP models, including our knowledge graph-based fact checking classifier and its execution on textual data containing SNSW. By providing a transformed version of the text in standard English, the SNSW translator enhances the compatibility and effectiveness of NLP models. This opens up new possibilities for improved prediction performance not just in fact checking, but also in various NLP tasks such as sentiment analysis, POS tagging, and machine translation. These improvements can potentially benefit various fields such as financial analysis, healthcare, legal discourse, or any industry that relies on specialized language and jargon. Additionally, translated articles can aid reviewers, policymakers, lawyers, and individuals by enabling better understanding of content containing SNSW, making it more accessible and facilitating informed decision making.

Furthermore, the SNSW translator has been designed with a focus on seamless integration with the work of other researchers. Additionally, SNSW-DB is designed with SQLLite to make it easy to maintain and use minimal memory. Moreover, we have developed interface APIs that allow for easy integration of both the database and the translator module. This design ensures that the models can be easily incorporated into future projects as a comprehensive solution or as individual modules, enabling researchers to build upon our work.

We evaluate the SNSW-DB and SNSW translator in multiple steps:

1. Assessing the ability to identify SNSW and generate corresponding formal synonyms.
2. Evaluating the quality of translations for the entire article into standard English.
3. Examining the classification performance impact on two pre-trained LLMs for a NLP task with the output of the SNSW translator.
4. Assessing the prediction performance of our knowledge graph-based fact checking (KG-FC) model on Reddit using the SNSW translator.

5.3 Methodology

In this section, we begin by outlining the overall architecture of the fact checking model tailored for Reddit, using the knowledge graph-based approach and the SNSW translator. Next, we detail the process of creating the SNSW database. We conclude with an introduction to the methodology behind SNSW Translation.

[7]https://en.wiktionary.org/

5.3.1 Overall Architecture of the Fact Checking Model

Figure 5.1 illustrates the high-level architecture of our knowledge graph-based fact checking model tailored for Reddit. When presented with a new claim C, the model employs a systematic approach involving two steps to process the claim and subsequently make a prediction. These steps are delineated as follows:

1. *Claim translation:* Initially, the SNSW translator is used to convert the claim's textual content into standard English, ensuring the removal of any SNSW terms.

2. *Knowledge graph-based fact checking model:* The translated text is then channeled through the knowledge graph-based fact checking model for classification.

The following sections will provide detailed descriptions of each individual module in our model, shedding light on their inner workings and the techniques.

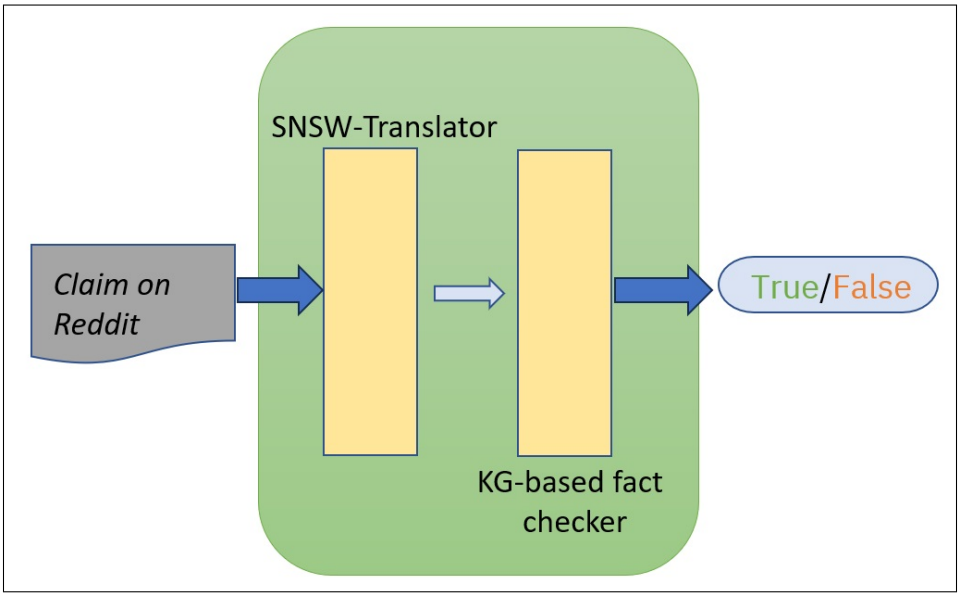


Figure 5.1: Fact checking model on Reddit

5.3.2 Methodology for SNSW Database Creation

This section describes the methodology for the creation of the database containing SNSW and their standard English synonyms. Following are the steps involved:

- Collecting SNSW
- Pre-processing of the collected data

- Generating standard English synonyms for SNSW
- Creating a database of SNSW, standard English synonyms, definitions, and example sentences.

Data Collection

To gather data on SNSW, we sourced online dictionaries that have gained widespread usage and feature substantial collections of user-generated content dating back 24 years. Specifically, we relied on Urban Dictionary and Wiktionary for this purpose.

Urban Dictionary: We used web scraping to extract SNSW from Urban Dictionary, using the Python library BeautifulSoup [133]. We then used the Urban Dictionary APIs to retrieve relevant data for each extracted word. For each entry, we collected the word/phrase, definition, example sentences, and upvote and downvote counts (with the votes provided by other users). For each word, we only store the entry corresponding to the one with the highest upvote count.

Wiktionary: Wiktionary is a multilingual online dictionary that contains definitions and translations of words and phrases in many languages. We used the Wiktionary APIs to extract data from Wiktionary. For each entry of Urban Dictionary we collected the corresponding entry from Wiktionary, if it exists. For each such entry, we collected the word/phrase, the grammatical category of the term, definition and example sentences.

Overall, our data collection process allowed us to gather a large volume of SNSW from two popular online dictionaries. The result is a list of SNSW along with their various details, including a substantial amount of noisy data such as spam words, made up words, and standard English words.

Data Pre-processing

Urban Dictionary is a crowdsourced platform, and not all the entries are reliable or accurate. For example it contains simple spam words like '123456', 'AAAAAA', 'AABBBBBYYYYYYYYY', and other words such as names of persons and institutions, standard English words, and made up words. Thus, it is crucial to filter out unnecessary terminologies. To achieve this, we referred to the study [134] which provides statistical properties of entries in Urban Dictionary, based on which we implement the following basic filter pipeline to identify and remove simple spam words.

1. Remove numeric words, e.g., '123456'
2. Remove words in which all characters are the same., e.g., 'AAAAAA'
3. Remove words with repeated patterns, e.g., 'ABCABC'

Then, we use an advanced filter pipeline for filtering complex spam words as follows,

1. Filter words with wScore less than 200. Where wScore is defined in Equation 5.1, and counts represents votes provided by other users.

$$wScore = |upVoteCount - downVoteCount| \tag{5.1}$$

2. To remove spam definitions for famous persons and places, we apply a named entity recognition filter on the example sentence and remove the identified entities.

3. There also exists standard words and their definition. Based on our definition of SNSW, these words do not qualify for the study. Thus, for filtering these terms we use a corpus of standard words [135]. However, some standard words may have multiple meanings, including a more colloquial meaning that would be relevant for our purposes. To account for these instances, we add a collection of common SNSW [136].

4. Finally, if the word is present in Wiktionary, then we use its definition; otherwise, we use the definition obtained from Urban Dictionary. This is done as Wiktionary, being maintained by the Wikimedia Foundation, is more reliable, controlled, and less prone to error than Urban Dictionary.

After these steps, we obtain a list of 21,570 SNSW ($listSNSW$) containing words/ abbreviations/phrases along with their corresponding examples and meaning.

Synonym Generation

We developed a synonym generator to identify standard English words which are equivalent to their SNSW counterparts. In various applications, such as search engines, content generation, and text classification, where a diverse and extensive vocabulary is essential, mapping SNSW to standard synonyms proves highly advantageous. Furthermore, downstream NLP applications that prioritize context preservation can greatly benefit from this mapping, as by merely marking and removing SNSW from text, the user's contextual metadata decreases. However, our model replaces these terms with their standard English counterparts, thus preserving the user's context and benefiting downstream NLP applications.

To begin, we randomly sampled the well-known WordNet dataset [137] to compile a list of 2000 common standard English words and their definitions from Wiktionary, which we refer to as $formalDB$. WordNet is a lexical database and semantic network of English words. It is a large, organized collection of words, their meanings, and related concepts. To improve the diversity and uniqueness of the synonym generation process, we retained a word and removed its synonyms from $formalDB$, resulting in 1824 words. This allowed us to obtain a more varied range of synonyms for each word in SNSW-DB.

To get the standard synonyms for each SNSW, firstly we use Distill-Bert to get the embeddings of the definitions of words in $formalDB$ and $listSNSW$. Afterward, for each word in $listSNSW$, we obtained the corresponding entry in $formalDB$ that has the minimum distance using cosine as the dissimilarity metric. The method is illustrated in Algorithm 1.

Algorithm 1 Synonym generation algorithm

Require: $listSNSW, formalDB$
  $y \leftarrow 1$
  $X \leftarrow x$
  $N \leftarrow n$
  while $fWord \leftarrow formalDB$ **do**
    $def \leftarrow fWord_{definition}$
    $embb \leftarrow getEmbedding(def)$ ▷ Using Distill-Bert
    $fWord_{embedding} \leftarrow embb$
  end while
  while $sWord \leftarrow listSNSW$ **do**
    $def \leftarrow sWord_{definition}$
    $embb \leftarrow getEmbedding(def)$ ▷ Using Distill-Bert
    $sWord_{embedding} \leftarrow embb$
    $fWord \leftarrow minCosineDist(embb, formalDB)$
    $sWord_{synonym} \leftarrow fWord_{word}$
  end while

SNSW Database Creation

This module creates the database (SNSW-DB) which contains the following entries:

- SNSW (words, phrases, abbreviations)
- Definitions of the SNSW
- Example sentences containing SNSW
- Corresponding standard English synonyms

This database can be used independently for identifying and removing/replacing SNSW in an article. It can also be used as part of the SNSW translator module as described in Section 5.3.3. We store SNSW-DB in SQLite, with a cache size of 128 keys for improved functionality and scalability. SQLite is a file-based DBMS that is known for its simplicity, reliability, and flexibility. It is also a serverless, self-contained, and zero-configuration DBMS, which makes it easy to integrate our work with other applications. Additionally, we added interface APIs for our modules. This allows SNSW-DB to be easily used in research projects as well as mobile and web applications.

5.3.3 Methodology for SNSW Translation

The architecture of the SNSW translator is shown in Figure 5.2. It consists of two main modules: the SNSW replacer module and translation module. For the first module, we use SNSW-DB, for its collection of SNSW and the corresponding formal synonyms.

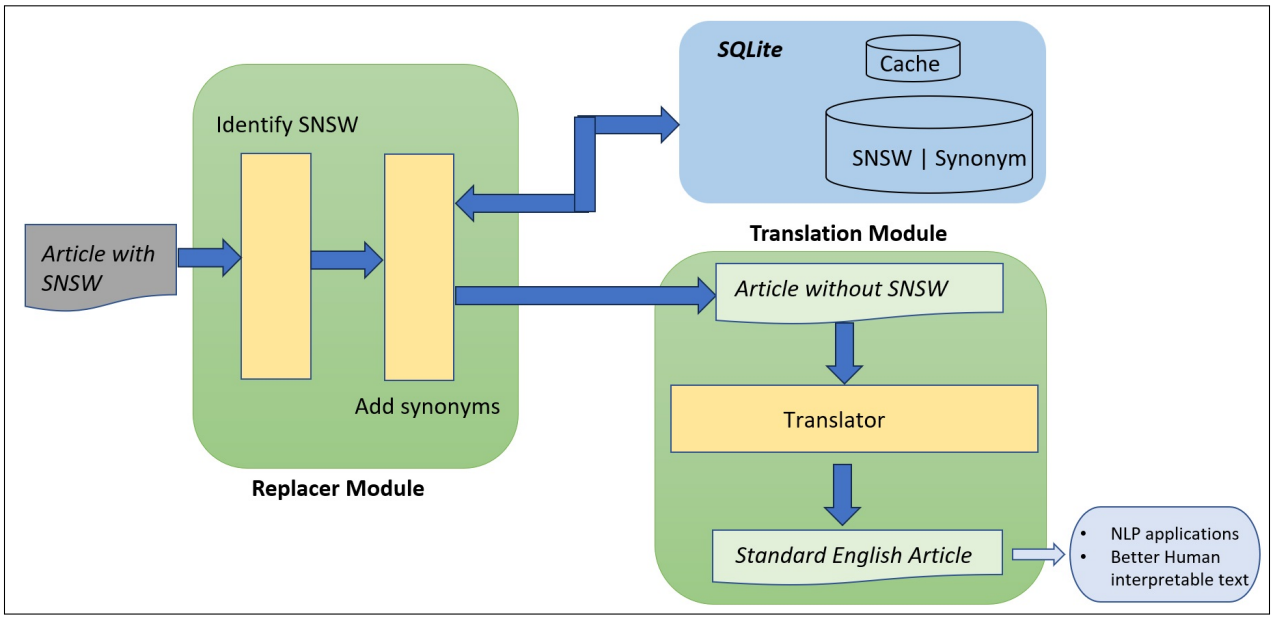


Figure 5.2: SNSW replacer and article translator

Replacer Module

We tokenize each article and check if any words or phrases are present in the SNSW-DB. If they are, we flag them as SNSW. While we do have the capability to delete the flagged words, doing so reduces the user context. Instead, we replace each flagged word with its corresponding standard English synonym from the SNSW-DB, which maintains the user context. Nevertheless, the dictionary substitution step can lead to incorrect grammar, which may make the article difficult to process for downstream NLP applications as well as for humans to understand. This is where the article translator module comes into play, as explained in the next section.

Article Translator

The conversion of articles containing standard English synonyms of SNSW (the machine-generated output text from the previous Replacer module) to human readable content is a challenging task that can be approached as a machine translation problem. In this study, we define this as a problem of converting 'broken English' to 'correct English' using machine generated translation, specifically addressing the Grammatical Error Correction (GEC) task. The objective is to translate 'broken English', which may include grammatical errors, sentence fragments, or other non-standard language usage, into standard English that is understandable to a human reader and can be used as input to downstream NLP models. Previous research has explored similar problems, such as the conversion of SMS text to standard English [138].

In this study, we use transfer learning to fine-tune the text-to-text T5-base model released by Google AI. T5 is a state-of-the-art transformer model and excels at sequence-to-sequence modeling tasks such as machine translation, summarization, and text generation [139]. The T5 algorithm is trained on a massive amount of textual data. We fine-tuned the model using the JFLEG dataset [140], which is often used for GEC task, and used Adam [141] as the optimizer. However, to tailor the dataset to our specific problem, we modified it by randomly lemmatizing words with a probability $p = 0.05$. This ensured the dataset's relevance and adaptability to our particular research context.

Table 5.1: Translator model loss

Step	**Loss**
Before training	1.64
After training	0.52

To evaluate the model's performance, we employed the loss metric, which measures the disparity between the model's predictions and the correct answers. From Table 5.1 we can observe that the loss decreased significantly after the training. Table 5.2 displays the hyper-parameters used for the model.

Table 5.2: Hyper-parameters for translator model

Parameter	**Value**
Learning rate	3e-4
Train epochs	20
Batch size	10

The output of this module is standard English which is both grammatically and syntactically correct. Since most common word embedding models are not trained on corpus containing SNSW, they may generate inaccurate embeddings for such words, which could impact the overall classification performance of NLP tasks. However, the resulting article of this module is free of SNSW and can be used for downstream NLP tasks. Furthermore, this model holds broad applications across fields such as the legal industry, language and communication studies, and digital content creation. Professionals from different fields can use the resulting articles to improve their work in various ways. For example, lawyers can use the articles to gain a better understanding of legal texts and cases, while psychologists can use it to analyze communication patterns and identify trends. Linguists can use the model to study language usage, and marketers can use it to analyze customer feedback and develop effective marketing strategies. Journalists can use the model to analyze articles and develop better reporting.

5.3.4 Sample Translation

Figure 5.3 displays a sample generated by our model. The initial text was analyzed to identify the presence of SNSW, after which synonym substitutions were made. The substituted text was then translated into standard English. This output can be used by downstream NLP applications and humans.

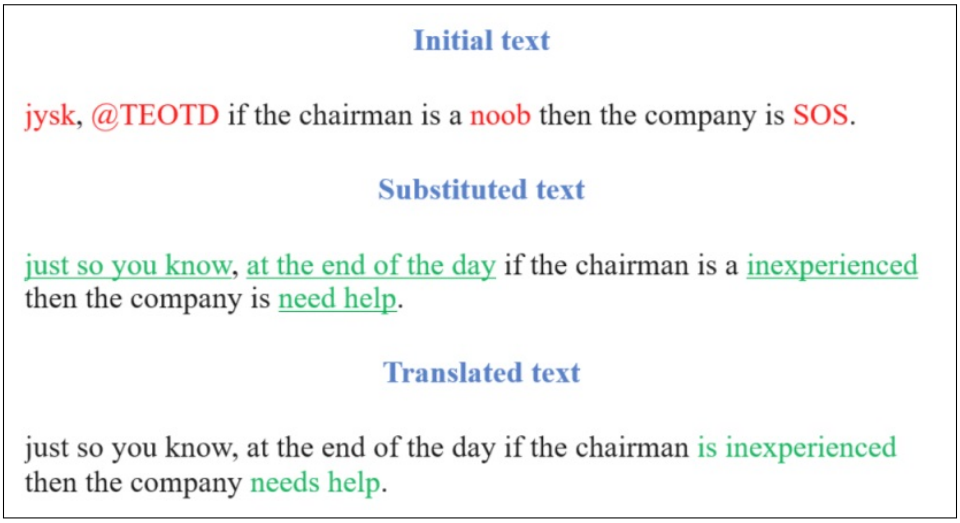


Figure 5.3: Sample from SNSW translator with SNSW terms in red

5.4 Datasets

5.4.1 Data Preparation

For fact checking on Reddit, we used the popular 'r/Fakeddit' dataset [142]. The Fakeddit dataset is a multi-modal benchmark dataset for fine-grained fake news detection, consisting of over 1 million samples from multiple categories of fake news collected from Reddit, making it one of the most extensive publicly available datasets. The dataset is collected in 2019. The dataset provides text and image for each sample, as well as labels for 2-way (false/true), 3-way (false/partly-true/true), and 6-way (satire, hoax, propaganda, trusted, false connection, and misleading) classification categories. For this study, we considered the 2-way labels.

There are a total of 628,501 false samples and 527,049 true samples in the Fakeddit dataset. Resulting in a balanced 54:46 class distribution. The dataset is split into 80:20 for training and testing.

5.5 Evaluation

First, we evaluate the effectiveness of SNSW-DB as a SNSW identifier. Second, we evaluate the generated SNSW translations and measure the classification performance impact of two pre-trained deep learning models for a NLP task with and without the output of our model. Finally, we evaluate our knowledge graph-based model with the output of the SNSW translator, for fact checking on Reddit.

5.5.1 Metrics

Let TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively. We employed four evaluation criteria extensively used in text classification tasks, namely accuracy, precision, recall and F1 score, to evaluate the prediction performance of the models. Additionally, we employed the bilingual evaluation understudy (BLEU) metric to evaluate the SNSW translator:

- Accuracy (A): Accuracy is a measure of the classifier's ability to correctly classify a piece of claim as either true or false.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5.2}$$

- Precision (P): Precision is a measure for the classifier's exactness such that a low value indicates a large number of false positives. The precision represents the number of positive predictions divided by the total number of positive class values predicted.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.3}$$

- Recall (R): Recall is considered a measure of a classifier's completeness (e.g., a low value of recall indicates many false negatives), where the number of true positives is divided by the number of true positives and the number of false negatives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.4}$$

- F1 score (F1): The F1 score is calculated as the weighted harmonic mean of the precision and recall measures of the classifier.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.5}$$

- Bilingual evaluation understudy (BLEU): BLEU measures the similarity between candidate translations and reference translations based on n-gram precision, offering a numerical assessment of translation quality. Scores range from 0 (indicating no match)

to 1 (indicating a perfect match). Let p_n is the precision for n-grams, w_n are the weights for each n-gram (often set such that $\sum w_n = 1$), c being the length of the candidate translation and r is the effective reference length. The BLEU score is calculated as:

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{5.6}$$

and BP is the brevity penalty, given by:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases}$$

5.5.2 Results and Discussion

All experiments are conducted on a machine with Intel(R) Core(TM) i7-8750H 2.20GHz CPU, 16 GB of RAM and NVIDIA GeForce RTX 4090 GPU running Windows 11. For the knowledge graph-based classifier, the ConvE models used the same hyper-parameters as discussed in Section 3.5.3 of Chapter 3.

SNSW Identification in Articles

For evaluating the effectiveness of the SNSW-DB as a SNSW identifier, we use the "Wikipedia" dataset published by Alphabet's Conversation AI team, which contains comments from Wikipedia's talk page edits [143]. This dataset contains 127,820 comments from Wikipedia talk pages labeled with whether or not they contain slang words. We compare the SNSW-DB against the following three state-of-the-art models,

1. profanity-check [144]
2. profanity-filter [145]
3. profanity [146]

Where, the first two are data driven model and the last one is a dictionary based model. Table 5.3 provides an overview of the time taken (in milliseconds) by various models to identify the first SNSW, ten SNSW, and 100 SNSW. This comparison evaluates the performance of different models in identifying SNSW and their scalability for real-world applications. Our model using SNSW-DB surpasses the other dictionary based model in terms of performance, which can be attributed to the utilization of the SQLLite database management system, enhancing the resolution speed of our model. Additionally, our model has comparable speed to the SVM based model "profanity-check".

Moreover, Table 5.4 indicates that our model boasts the highest accuracy. This can be attributed to the use of the exhaustive SNSW-DB. While the "profanity-filter" is also highly accurate, it compromises performance. In contrast, the "profanity-check" is fast but has lower accuracy than our model. Thus, compared to the baseline models our SNSW identifier model has higher accuracy and comparable time consumption.

Table 5.3: Comparison of running time of models in milliseconds

Model	1 Prediction	10 Predictions	100 Predictions
profanity-check [144]	0.2	**0.5**	**3.5**
profanity-filter [145]	60	1200	13000
profanity [146]	0.3	1.2	24
SNSW translator	**0.1**	1.0	4.2

Table 5.4: Performance Comparison of the Models

Model	Test Accuracy	Precision	Recall	F1 Score
profanity-check [144]	0.950	0.861	0.896	0.878
profanity-filter [145]	0.918	0.854	0.702	0.771
profanity [146]	0.856	**0.917**	0.308	0.461
SNSW translator	**0.972**	0.876	**0.903**	**0.889**

Translating SNSW in Documents to Standard English

Our SNSW translator model was evaluated using widely adopted metric for evaluating machine translation tasks, namely bilingual evaluation understudy (BLEU) metric [147].

We curated a reference corpus by manually translating 100 sentences from the output of the 'Replacer Module'. These translations served as the reference for evaluation. The candidate corpus, on the other hand, was generated by the 'Article Translator Module', representing the machine generated translations that were being evaluated. We evaluated two instances of the SNSW translator, one translating abbreviations in text and the other ignoring abbreviations, as abbreviations add a layer of complexity. This way we were able to analyze the impact of handling abbreviations on the model's performance and translation quality.

Table 5.5: Performance scores for our SNSW translator using BLEU metric

Model	BLEU
SNSW translator using SNSW-DB (with abbreviations)	0.471
SNSW translator using SNSW-DB (without abbreviations)	0.523

A BLEU score between 0.3 and 0.6 is considered moderate to good quality translation. It can be observed from Table 5.5 that the BLEU scores for both experiments fall within an acceptable range. However, the model instance that does not consider abbreviations achieves a higher BLEU score, indicating better translations. This is likely due to the challenges associated with disambiguating abbreviations and the limitations of the BLEU metric in capturing the accuracy and fluency of translations involving abbreviations. Overall, these

results demonstrate the usefulness of our approach and highlight the potential for the SNSW translator to facilitate the translation of colloquial language.

Impact of the SNSW Translator on the Performance of NLP Tasks

To measure the impact on downstream NLP models, we consider sarcasm detection as the NLP task. We use a BERT-based model [148] and a RoBERTa-based model [149] for sarcasm detection from Hugging Face and GitHub, respectively. The authors of these models had fine-tuned them on news datasets that does not contain SNSW.

For evaluating the SNSW translator, we use the SARC dataset [150], which is used for sarcasm detection and comprises of comments from Reddit containing SNSW. Given that the SARC dataset encompasses 1.3 million samples, for the sake of simplicity, we randomly selected 10,000 samples, ensuring a balanced 50:50 class distribution between the sarcasm and non-sarcasm categories.

Table 5.6: Effect of model output on downstream sarcasm detection models

Model	**Accuracy**
BERT model	0.551
BERT model + SNSW translator	0.614
RoBERTa model	0.517
RoBERTa model + SNSW translator	0.568

In our study, we conducted two experiments: one with the output of the SNSW translator, and one without it. Table 5.6 demonstrates that the downstream BERT-based and RoBERTa-based model's accuracy was increased by 6.24% points and 5.15% points respectively. These improvements can be attributed to the output of the SNSW translator. Specifically, each input sentence is first passed through the SNSW translator, which identifies SNSW and then translates the entire sentence into standard English, aligning it with the vocabulary on which the NLP models were trained. This process allowed the NLP models to better understand and process the sentence, leading to the observed increase in accuracy.

Fact Checking on Reddit Using KG-FC

Table 5.7 presents the evaluation of our fact checking model and also compares it to baseline models.

We compared our proposed model with the following six state-of-the-art baseline misinformation detection models,

1. att-RNN [151]: This model adopts attention module to identify fake news based on the textual, visual and social context features. Image features are incorporated into the joint features of text and social context, which are obtained with an LSTM network, to produce the fused classification model.

2. NeuralTalk [152]: NeuralTalk is proposed for image caption by using deep recurrent framework. the authors obtained news representations by averaging the output of the RNN at each time step, which are then fed into a fully connected layer followed by an entropy loss layer to make predictions.

3. VGG 19 + Text-CNN [153]: The authors used multi-modal classifier ensemble to fuse the outputs from Text-CNN with VGG-19 for fake news detection. Text-CNN applies convolutional neural network for classification by using multiple filters with different sizes to extract the key information. The authors also evaluated [151] and [152].

4. MVAE [154]:This model contains a variational autoencoder which is capable of learning probabilistic latent variable models by optimizing a bound on the marginal likelihood of the observed data.

5. DeBERTNeXT [155]: This model usees both textual and visual data from articles fake news detection. The authors used a fusion of DeBERTa model and the ConvNeXT models. The authors also evaluated MVAE [154] on the fakeddit dataset using the same set of hyperparameters as DeBERTNeXT.

6. BERT + ResNet50 [119]: The authors of the dataset trained and tested BERT for text and ResNet50 for the images on the dataset for fake news detection.

It can be inferred from Table 5.7 that the proposed knowledge graph-based classifier with SNSW translator outperforms all the state-of-the-art models in terms of accuracy while attaining high precision and recall. In particular, out model achieves an accuracy of 91.474%. While the DeBERTNeXT model registers a comparable accuracy, it is essential to highlight that DeBERTNeXT uses both textual and image data for its classification. In contrast, our model solely uses the textual data offering an advantage as image data might not always be available.

Additionally, the proposed knowledge graph-based model without the SNSW translator achieved an accuracy of 86.167%, whereas with the output of the translator in the pipeline the accuracy increases by 5.3%. This improvement can be ascribed to the presence of SNSW in the articles, which often refers to specific relations and entities. Translating these SNSW to standard English is important for the thorough extraction of all the triplets.

Table 5.7: Comparison with baseline models

Model	Accuracy	Precision	Recall	F1 Score
att-RNN [151]	0.745	0.712	0.675	0.693
NeuralTalk [152]	0.612	0.655	0.661	0.658
VGG 19 + Text-CNN [153]	0.804	0.771	0.738	0.754
MVAE [154]	0.784	- -	- -	- -
DeBERTNeXT [155]	0.912	**0.913**	0.902	**0.907**
BERT + ResNet50 [119]	0.891	- -	- -	- -
Our KG-FC model	0.861	0.875	0.885	0.880
Our KG-FC model with SNSW translator	**0.914**	0.897	**0.917**	**0.907**

5.6 Limitations of the Model

The following are the limitations of our models:

- One limitation our model faces is the evolution of SNSW meanings over time. Despite regular updates, newly introduced or altered SNSW between the updates remain unaccounted for.
- Some content on platforms like Reddit includes not just text, but images, videos, and hyperlinks. Extracting knowledge or context from such multi-modal content is challenging.

5.7 Chapter Summary

Misinformation identification is pivotal for platforms like Reddit and Twitter. However, on platforms like Reddit, the pervasive use of slang and non-standard words (SNSW) complicates traditional fact checking methods as SNSW can decrease the accuracy of NLP based models. In response, we propose a SNSW translator that converts SNSW to standard English. The output from the translator is input into the KG-based fact checking (KG-FC) model. The standard English output significantly improves the classification performance of the KG-FC model as demostrated by our experiment results.

'Translated' text not only enhances the accuracy of our classifier, but also increases the prediction performance of other NLP tasks like sentiment analysis and improves reading comprehension by humans.

This study can be extended in the following directions:

- Augmenting our initial SNSW dataset with additional crowdsourced dictionaries to broaden the terminology coverage.

- Investigating the feasibility of real-time fact-checking for live events or trending topics, with potential impacts on breaking news or viral content dissemination.
- Extending the model to address SNSW and fact-checking challenges in languages other than English, enhancing the tool's reach and relevance due to the global nature of many platforms.
- Extending this study for fact-checking on other online social networks where SNSW pose a challenge, such as Facebook and Threads.

Chapter 6

Conclusion and Future Work

In this chapter, we summarize the main results of the thesis, identify open issues, and outline research directions for future work.

6.1 Summary

The digital age, characterized by rapid information dissemination, has underscored the importance of credible and reliable news content. Misinformation has evolved into a multifaceted challenge threatening the economy, democracy, social stability, public health, and national security. Our research in this thesis aims to bridge the gap between traditional fact checking methods and the exponential growth of digital content.

This thesis comprises several key areas of research, each contributing uniquely to the topic of misinformation detection. The contributions are:

- We propose a fact checking model that uses two separate knowledge graphs (KG), one containing true claims and the other, false claims (misinformation). We use ConvE, a CNN-based knowledge graph embedding method, to train the model. Thanks to the addition of the false claim KG, the proposed model achieves higher accuracy when compared with the case of using only the true claim KG. Experimental results show that the proposed model outperforms the baseline misinformation detection models in terms of classification accuracy. The predictions are also accompanied by XAI output, which reduces cost of errors, and enables end-users to understand the rationales behind the model's predictions.

- We propose a machine learning model using diffusion cascades of tweets on Twitter to determine if a post is likely to be misinformation or not. The propagation-based model also uses "infectiousness" properties of misinformation, which are learned from the spread of COVID-19 and epidemiological models of infectious diseases. The propagation-based model is combined with the above KG-based model, achieving higher accuracy when compared with the case of using either model independently. In addition, we provide

explanations using the diffusion pattern of a tweet to show the the rationales behind the propagation-based model's prediction.

- We propose a transformer-based model that converts text with slang and non-standard words (SNSW) into standard English for misinformation detection on Reddit. The output in standard English is then input into the above KG-based fact-checking model. The 'translated' text increases the classification accuracy of the model in comparison to the case where the input includes SNSW. Furthermore, our experimental results demonstrate that the use of 'translated' text by other downstream NLP tasks (e.g., sentiment analysis) leads to improved prediction performance.

The dynamic nature of misinformation and the ever-evolving digital landscape present continuous challenges that require ongoing adaptation and enhancement of our models. While our research represents a step forward in misinformation detection, it has its own limitations, as follows.

- The models heavily depends on the quality and completeness of the underlying knowledge graphs. Inaccurate or biased claims within these graphs could lead to erroneous predictions.
- The interpretability offered by XAI techniques could be further enhanced by incorporating additional information, such as the credibility of sources and the origins of the information.
- The propagation-based classifier may not generalize well to other social media platforms, such as Instagram and Facebook, due to their distinct network structures.
- The models presented in this work are designed to assist fact checkers, but the ultimate judgment on the veracity of claims still rests with human users, who may introduce their own subjectivity and bias.

6.2 Future Work

Following are several potential research directions for future work:

- Widening the scope of evaluation by testing the proposed models on diverse datasets, particularly from different domains, to understand its universal applicability.
- Investigating further into XAI techniques to provide more detailed and easy-to-understand explanations.
- Investigating the feasibility of real-time fact checking for live events or trending topics, which could have significant implications for breaking news or viral content dissemination.

- Exploring alternative types of neural networks for knowledge graph embeddings to build fact checking models.
- Extending the study for fact checking on other online social networks, such as TikTok, Pinterest and Facebook.

Bibliography

[1] Andrew M. Guess and Benjamin A. Lyons. "Misinformation, Disinformation, and Online Propaganda". In: *Social Media and Democracy: The State of the Field, Prospects for Reform*. Ed. by Nathaniel Persily and Joshua A.Editors Tucker. SSRC Anxieties of Democracy. Cambridge University Press, 2020, pp. 10–33.

[2] Hunt Allcott and Matthew Gentzkow. "Social Media and Fake News in the 2016 Election". In: *Journal of Economic Perspectives* 31.2 (May 2017), pp. 211–36. DOI: 10.1257/jep.31.2.211.

[3] Stephan Lewandowsky et al. "Misinformation and Its Correction: Continued Influence and Successful Debiasing". In: *Psychological Science in the Public Interest* 13.3 (2012), pp. 106–131. ISSN: 15291006.

[4] A. Coleman. *'Hundreds dead' because of covid-19 misinformation*. Accessed: 2021-07-15. 2020. URL: https://www.bbc.com/news/world-53755067.

[5] G. Pennycook and D. G. Rand. "Fighting misinformation on social media using crowdsourced judgments of news source quality". In: *Proceedings of the National Academy of Sciences of the United States of America* 116.7 (2019), pp. 2521–2526. DOI: 10.1073/pnas.1806781116.

[6] Kai Shu et al. "Fake News Detection on Social Media: A Data Mining Perspective". In: *SIGKDD Explor. Newsl.* 19.1 (Sept. 2017), pp. 22–36. ISSN: 1931-0145. DOI: 10.1145/3137597.3137600.

[7] Xinyi Zhou and Reza Zafarani. "A Survey of Fake News". In: *ACM Computing Surveys* 53.5 (Sept. 2020), pp. 1–40. DOI: 10.1145/3395046.

[8] Giovanni Luca Ciampaglia et al. "Computational Fact Checking from Knowledge Networks". In: *PLOS ONE* 10.6 (June 2015), pp. 1–13. DOI: 10.1371/journal.pone.0128193.

[9] Yuanfei Dai et al. "A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks". In: *Electronics* 9.5 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9050750.

[10] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. "A Survey on Automated Fact-Checking". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 178–206. DOI: 10.1162/tacl_a_00454.

[11] James Thorne et al. "FEVER: a Large-scale Dataset for Fact Extraction and VERification". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 809–819. DOI: 10.18653/v1/N18-1074.

[12] Ling Min Serena Khoo et al. "Interpretable rumor detection in microblogs by attending to user interactions". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05. 2020, pp. 8783–8790.

[13] Andreas Vlachos and Sebastian Riedel. "Fact Checking: Task definition and dataset construction". In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics, June 2014, pp. 18–22. DOI: 10.3115/v1/W14-2508.

[14] Mykhailo Granik and Volodymyr Mesyura. "Fake news detection using naive Bayes classifier". In: *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. 2017, pp. 900–903. DOI: 10.1109/UKRCON.2017.8100379.

[15] Arkaitz Zubiaga et al. "Detection and Resolution of Rumours in Social Media: A Survey". In: *ACM Comput. Surv.* 51.2 (Feb. 2018). ISSN: 0360-0300. DOI: 10.1145/3161603.

[16] Hannah Rashkin et al. "Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2931–2937. DOI: 10.18653/v1/D17-1317.

[17] Benjamin Riedel et al. *A simple but tough-to-beat baseline for the Fake News Challenge stance detection task*. 2018. arXiv: 1707.03264 [cs.CL].

[18] William Ferreira and Andreas Vlachos. "Emergent: a novel data-set for stance classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL. 2016.

[19] Ali K Chaudhry, Darren Baker, and Philipp Thun-Hohenstein. "Stance detection for the fake news challenge: identifying textual relationships with deep neural nets". In: *CS224n: Natural Language Processing with Deep Learning* (2017), pp. 1–117.

[20] Pallabi Saikia et al. *Modelling Social Context for Fake News Detection: A Graph Neural Network Based Approach*. 2022. arXiv: 2207.13500 [cs.SI].

[21] Kenan Xiao et al. "Looking Beyond Content: Modeling and Detection of Fake News from a Social Context Perspective". In: *55th Hawaii International Conference on System Sciences, HICSS 2022, Virtual Event / Maui, Hawaii, USA, January 4-7, 2022*. ScholarSpace, 2022, pp. 1–10.

[22] Sejeong Kwon et al. "Prominent Features of Rumor Propagation in Online Social Media". In: *2013 IEEE 13th International Conference on Data Mining*. 2013, pp. 1103–1108. DOI: 10.1109/ICDM.2013.61.

[23] Md Bhuiyan et al. "NudgeCred: Supporting News Credibility Assessment on Social Media Through Nudges". In: *Proceedings of the ACM on Human-Computer Interaction* 5 (Oct. 2021), pp. 1–30. DOI: 10.1145/3479571.

[24] Vasu Agarwal et al. "Analysis of Classifiers for Fake News Detection". In: *Procedia Computer Science* 165 (2019). 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019, pp. 377–383. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.01.035.

[25] Arup Baruah et al. "Automatic Detection of Fake News Spreaders Using BERT". In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*. Ed. by Linda Cappellato et al. Vol. 2696. CEUR Workshop Proceedings. CEUR-WS.org, 2020.

[26] Yong Fang et al. "Self Multi-Head Attention-based Convolutional Neural Networks for fake news detection". In: *PLOS ONE* 14.9 (Sept. 2019), pp. 1–13. DOI: 10.1371/journal.pone.0222713.

[27] R. K. Kaliyar, A. Goswami, and P. Narang. "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach". In: *Multimedia Tools and Applications* 80.8 (2021), pp. 11765–11788. DOI: 10.1007/s11042-020-10183-2.

[28] M. Mayank, S. Sharma, and R. Sharma. "DEAP-FAKED: Knowledge Graph based Approach for Fake News Detection". In: *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2022, pp. 47–51. DOI: 10.1109/ASONAM55673.2022.10068653.

[29] Isabelle Augenstein et al. "MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4685–4697. DOI: 10.18653/v1/D19-1475.

[30] Yudian Zheng et al. "Truth inference in crowdsourcing: Is the problem solved?" In: *Proceedings of the VLDB Endowment* 10.5 (2017), pp. 541–552.

[31] Nayeon Lee et al. "Towards Few-shot Fact-Checking via Perplexity". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 1971–1981. DOI: 10.18653/v1/2021.naacl-main.158.

[32] Mourad Sarrouti et al. "Evidence-based Fact-Checking of Health-related Claims". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3499–3512. DOI: 10.18653/v1/2021.findings-emnlp.297.

[33] Wenxuan Zhang et al. "AnswerFact: Fact Checking in Product Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2407–2417. DOI: 10.18653/v1/2020.emnlp-main.188.

[34] Eric Lazarski, Mahmood Al-Khassaweneh, and Cynthia Howard. "Using NLP for Fact Checking: A Survey". In: *Designs* 5.3 (2021). ISSN: 2411-9660. DOI: 10.3390/designs5030042.

[35] Miguel A. Alonso et al. "Sentiment Analysis for Fake News Detection". In: *Electronics* 10.11 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10111348.

[36] Nguyen Vo and Kyumin Lee. "Learning from Fact-Checkers: Analysis and Generation of Fact-Checking Language". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'19. Paris, France: Association for Computing Machinery, 2019, pp. 335–344. ISBN: 9781450361729. DOI: 10.1145/3331184.3331248.

[37] Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. *Reading and Reasoning over Chart Images for Evidence-based Automated Fact-Checking*. 2023. arXiv: 2301.11843 [cs.CL].

[38] Chris Samarinas, Wynne Hsu, and Mong Li Lee. "Improving Evidence Retrieval for Automated Explainable Fact-Checking". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*. Online: Association for Computational Linguistics, June 2021, pp. 84–91. DOI: 10.18653/v1/2021.naacl-demos.10.

[39] Baoxu Shi and Tim Weninger. "Discriminative predicate path mining for fact checking in knowledge graphs". In: *Knowledge-Based Systems* 104 (2016), pp. 123–133. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2016.04.015.

[40] Peng Lin et al. "Discovering Patterns for Fact Checking in Knowledge Graphs". In: *Journal of Data and Information Quality* 11 (May 2019), pp. 1–27. DOI: 10.1145/3286488.

[41] Prashant Shiralkar et al. *Finding Streams in Knowledge Graphs to Support Fact Checking*. 2017. arXiv: 1708.07239 [cs.AI].

[42] Jeff Z. Pan et al. "Content Based Fake News Detection Using Knowledge Graphs". In: *International Workshop on the Semantic Web*. 2018.

[43] Antoine Bordes et al. "Translating Embeddings for Modeling Multi-Relational Data". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 2787–2795.

[44] Xiwei Xu and Ke Qin. "Interpretable Fake News Detection on Social Media". In: *Proceedings of the 2023 6th International Conference on Software Engineering and Information Management*. ICSIM '23. Palmerston North, New Zealand: Association for Computing Machinery, 2023, pp. 286–291. ISBN: 9781450398237. DOI: 10.1145/3584871.3584913.

[45] Cheng-Jun Yang Shih-Yi Chien and Fang Yu. "XFlag: Explainable Fake News Detection Model on Social Media". In: *International Journal of Human–Computer Interaction* 38.18-20 (2022), pp. 1808–1827. DOI: 10.1080/10447318.2022.2062113.

[46] Ken MISHIMA and Hayato YAMANA. "A Survey on Explainable Fake News Detection". In: *IEICE Transactions on Information and Systems* E105.D.7 (2022), pp. 1249–1257. DOI: 10.1587/transinf.2021EDR0003.

[47] Pew Research Center. *News Use Across Social Media Platforms in 2020*. Accessed: 2023-06-15. 2021. URL: https://www.pewresearch.org/journalism/2021/01/12/news-use-across-social-media-platforms-in-2020/.

[48] Soroush Vosoughi, Deb Roy, and Sinan Aral. "The spread of true and false news online". In: *Science* 359.6380 (2018), pp. 1146–1151. DOI: 10.1126/science.aap9559.

[49] Jonas L. Juul and Johan Ugander. "Comparing information diffusion mechanisms by matching on cascade size". In: *Proceedings of the National Academy of Sciences* 118.46 (2021), e2100786118. DOI: 10.1073/pnas.2100786118.

[50] Nir Rosenfeld, Aron Szanto, and David C. Parkes. "A Kernel of Truth: Determining Rumor Veracity on Twitter by Diffusion Pattern Alone". In: *Proceedings of The Web Conference 2020*. WWW '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 1018–1028. ISBN: 9781450370233. DOI: 10.1145/3366423.3380180.

[51] Colin Morris. *pejorative-compounds*. 2021. URL: https://github.com/colinmorris/pejorative-compounds (visited on 10/08/2023).

[52] JOHN AWA-ABUON. *Reddit Jargon Explained: Terms You Need to Know*. 2022. URL: https://www.makeuseof.com/reddit-jargon-explained-terms-you-need-to-know/ (visited on 10/08/2023).

[53] Katja Zupan, Nikola Ljubešić, and Tomaž Erjavec. "How to tag non-standard language: Normalisation versus domain adaptation for Slovene historical and user-generated texts". In: *Natural Language Engineering* 25.5 (2019), pp. 651–674. DOI: 10.1017/S1351324919000366.

[54] Peter Sheridan Dodds et al. "Human language reveals a universal positivity bias". In: *Proceedings of the National Academy of Sciences* 112.8 (2015), pp. 2389–2394.

[55] Alex Hern. "How social media filter bubbles and algorithms influence the election". In: *The Guardian* (May 2017). URL: https://www.theguardian.com/technology/2017/may/22/social-media-election-facebook-filter-bubbles.

[56] David Gunning and David Aha. "DARPA's Explainable Artificial Intelligence (XAI) Program". In: *AI Magazine* 40.2 (June 2019), pp. 44–58. DOI: 10.1609/aimag.v40i2.2850.

[57] European Commission. *"Commission appoints expert group on AI and launches the European AI alliance"*. https://ec.europa.eu/digital-single-market/en/news/commission-appoints-expert-group-ai-and-launches-european-ai-alliance. June 2018.

[58] Atul Rawal et al. "Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges, and Perspectives". In: *IEEE Transactions on Artificial Intelligence* 3.6 (2022), pp. 852–866. DOI: 10.1109/TAI.2021.3133846.

[59] Maria D. Molina et al. ""Fake News" Is Not Simply False Information: A Concept Explication and Taxonomy of Online Content". In: *American Behavioral Scientist* 65.2 (2021), pp. 180–212. DOI: 10.1177/0002764219878224.

[60] Jing Ma et al. "Detect Rumors Using Time Series of Social Context Information on Microblogging Websites". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM '15. Melbourne, Australia: Association for Computing Machinery, 2015, pp. 1751–1754. ISBN: 9781450337946. DOI: 10.1145/2806416.2806607.

[61] Adrien Friggeri et al. "Rumor Cascades". In: *Proceedings of the International AAAI Conference on Web and Social Media* 8.1 (May 2014), pp. 101–110. DOI: 10.1609/icwsm.v8i1.14559.

[62] Ke Wu, Song Yang, and Kenny Q. Zhu. "False rumors detection on Sina Weibo by propagation structures". In: *2015 IEEE 31st International Conference on Data Engineering*. 2015, pp. 651–662. DOI: 10.1109/ICDE.2015.7113322.

[63] Jing Ma, Wei Gao, and Kam-Fai Wong. "Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 708–717. DOI: 10.18653/v1/P17-1066.

[64] Chunyuan Yuan et al. "Early Detection of Fake News by Utilizing the Credibility of News, Publishers, and Users based on Weakly Supervised Learning". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5444–5454. DOI: 10.18653/v1/2020.coling-main.475.

[65] Shuo Yang et al. "Unsupervised Fake News Detection on Social Media: A Generative Approach". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 5644–5651. DOI: 10.1609/aaai.v33i01.33015644.

[66] Brendan Nyhan and Jason Reifler. "When Corrections Fail: The Persistence of Political Misperceptions". In: *Political Behavior* 32.2 (Mar. 2010), pp. 303–330. DOI: 10.1007/s11109-010-9112-2.

[67] Fan Yang et al. "XFake: Explainable Fake News Detector with Visualizations". In: *The World Wide Web Conference*. ACM, May 2019. DOI: 10.1145/3308558.3314119.

[68] Gordon Pennycook et al. "The implied truth effect: Attaching warnings to a subset of fake news headlines increases perceived accuracy of headlines without warnings". In: *Management science* 66.11 (2020), pp. 4944–4957.

[69] Ndapandula Nakashole and Tom M. Mitchell. "Language-Aware Truth Assessment of Fact Candidates". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 1009–1019. DOI: 10.3115/v1/P14-1095.

[70] Dean Pomerleau and Delip Rao. *Fake News Challenge*. http://fakenewschallenge.org/. Accessed: 2023-09-19. 2017.

[71] Andreas Hanselowski et al. *Description of the system developed by team Athene in the FNC-1*. https://github.com/hanselowski/athene_system/blob/master/system_description_athene.pdf. Accessed: 2023-03-13. 2017.

[72] Stephan Lewandowsky, Ullrich Ecker, and John Cook. "Beyond Misinformation: Understanding and Coping with the "Post-Truth" Era". In: *Journal of Applied Research in Memory and Cognition* 6 (July 2017). DOI: 10.1016/j.jarmac.2017.07.008.

[73] Shivani Choudhary et al. *A Survey of Knowledge Graph Embedding and Their Applications*. 2021. arXiv: 2107.07842 [cs.IR].

[74] Sina Mohseni et al. *Machine Learning Explanations to Prevent Overtrust in Fake News Detection*. 2020. arXiv: 2007.12358 [cs.IR].

[75] Benjamin D. Horne et al. "Rating Reliability and Bias in News Articles: Does AI Assistance Help Everyone?" In: *Proceedings of the International AAAI Conference on Web and Social Media* 13.01 (July 2019), pp. 247–256. DOI: 10.1609/icwsm.v13i01.3226.

[76] Mingxuan Chen, Ning Wang, and K. P. Subbalakshmi. *Explainable Rumor Detection using Inter and Intra-feature Attention Networks*. 2020. arXiv: 2007.11057 [cs.SI].

[77] Julio C S Reis et al. "Explainable Machine Learning for Fake News Detection". In: *Proceedings of the 10th ACM Conference on Web Science*. WebSci '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 17–26. ISBN: 9781450362023. DOI: 10.1145/3292522.3326027.

[78] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[79] Mengnan Du, Ninghao Liu, and Xia Hu. "Techniques for Interpretable Machine Learning". In: *Commun. ACM* 63.1 (Dec. 2019), pp. 68–77. ISSN: 0001-0782. DOI: 10.1145/3359786.

[80] Shaoxiong Ji et al. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (Feb. 2022), pp. 494–514. DOI: 10.1109/tnnls.2021.3070843.

[81] Bilal Abu-Salih. "Domain-specific knowledge graphs: A survey". In: *Journal of Network and Computer Applications* 185 (2021), p. 103076. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.103076.

[82] Mehdi Ali et al. "Bringing Light Into the Dark: A Large-Scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (Dec. 2022), pp. 8825–8845. DOI: 10.1109/tpami.2021.3124805.

[83] Quan Wang et al. "Knowledge Graph Embedding: A Survey of Approaches and Applications". In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743. DOI: 10.1109/TKDE.2017.2754499.

[84] Mojtaba Nayyeri et al. *Toward Understanding The Effect Of Loss function On Then Performance Of Knowledge Graph Embedding*. 2019. arXiv: 1909.00519 [cs.AI].

[85] Xiaojuan Tang et al. *RulE: Neural-Symbolic Knowledge Graph Reasoning with Rule Embedding*. 2023. arXiv: 2210.14905 [cs.AI].

[86] Tim Dettmers et al. "Convolutional 2D Knowledge Graph Embeddings". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press, 2018. ISBN: 978-1-57735-800-8.

[87] Shantanu Kumar. *A Survey of Deep Learning Methods for Relation Extraction*. 2017. arXiv: 1705.03645 [cs.CL].

[88] Maximilian Nickel et al. "A Review of Relational Machine Learning for Knowledge Graphs". In: *Proceedings of the IEEE* 104.1 (2016), pp. 11–33. DOI: 10.1109/JPROC.2015.2483592.

[89] Xin Dong et al. "Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: Association for Computing Machinery, 2014, pp. 601–610. ISBN: 9781450329569. DOI: 10.1145/2623330.2623623.

[90] PyKEEN Contributors. *Training PyKEEN 1.9.0 documentation*. https://pykeen.readthedocs.io/en/stable/reference/training.html. Accessed: 10-22-2023. 2023.

[91] Pere-Lluís Huguet Cabot and Roberto Navigli. "REBEL: Relation Extraction By End-to-end Language generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2370–2381. DOI: 10.18653/v1/2021.findings-emnlp.204.

[92] Tharun Sikhinam. *How does the performance of a graph database such as Neo4j compare to the performance of a relational database such as Postgres?* 2020. URL: https://courses.cs.washington.edu/courses/csed516/20au/projects/p06.pdf (visited on 09/19/2023).

[93] *TPC-H Homepage*. Transaction Processing Performance Council, 2021. URL: https://www.tpc.org/tpch/.

[94] *PostgreSQL: The World's Most Advanced Open Source Relational Database*. https://www.postgresql.org/. Accessed: 2023-07-09.

[95] William Yang Wang. ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 422–426. DOI: 10.18653/v1/P17-2067.

[96] Junaed Younus Khan et al. "A benchmark study of machine learning models for online fake news detection". In: *Machine Learning with Applications* 4 (June 2021), p. 100032. DOI: 10.1016/j.mlwa.2021.100032.

[97] Ashok Kumar Jayaraman, Tina Trueman, and Erik Cambria. "Fake News Detection Using XLNet Fine-Tuning Model". In: Nov. 2021, pp. 1–4. DOI: 10.1109/ICCICA52458.2021.9697269.

[98] Divyam Mehta et al. "A transformer-based architecture for fake news classification". In: *Social Network Analysis and Mining* 11 (Dec. 2021). DOI: 10.1007/s13278-021-00738-y.

[99] Nida Aslam et al. "Fake Detect: A Deep Learning Ensemble Model for Fake News Detection". In: *Complexity* 2021 (Apr. 2021), pp. 1–8. DOI: 10.1155/2021/5557784.

[100] Shaily Bhatt et al. "Fake News Detection: Experiments and Approaches Beyond Linguistic Features". In: *Data Management, Analytics and Innovation*. Springer Singapore, Sept. 2021, pp. 113–128. DOI: 10.1007/978-981-16-2937-2_9.

[101] Cass Sunstein. *Spread of false information causes dangers, says Sunstein*. 2008. (Visited on 09/27/2023).

[102] Arkaitz Zubiaga et al. "Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads". In: *PLOS ONE* 11.3 (Mar. 2016), pp. 1–29. DOI: 10.1371/journal.pone.0150989.

[103] Sinan Aral. *The Hype Machine: How Social Media Disrupts Our Elections, Our Economy, and Our Health*. New York, NY: Crown Currency, Sept. 2021.

[104] A. Heath. *Read the new Twitter CEO's first email to employees*. Accessed: 2023-09-15. 2023. URL: https://www.theverge.com/2023/6/12/23758258/twitter-ceo-linda-yaccarino-first-employee-memo.

[105] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. "Early vs Late Fusion in Multimodal Convolutional Neural Networks". In: *23rd International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–6. DOI: 10.23919/FUSION45008.2020.9190246.

[106] Becker NG et al. "Using Mathematical Models to Assess Responses to an Outbreak of an Emerged Viral Respiratory Disease". In: *National Centre for Epidemiology and Population Health* (Apr. 2006). DOI: ISBN1-74186-357-0.

[107] Julii Brainard et al. "Super-spreaders of novel coronaviruses that cause SARS, MERS and COVID-19: a systematic review". In: *Annals of Epidemiology* 82 (2023), 66–76.e6. ISSN: 1047-2797.

[108] Yuri Tani Utsunomiya et al. "Growth rate and acceleration analysis of the COVID-19 pandemic reveals the effect of public health measures in real time". In: *medRxiv* (2020). DOI: 10.1101/2020.03.30.20047688.

[109] Giovanni Minarini. "Root Mean Square of the Successive Differences as Marker of the Parasympathetic System and Difference in the Outcome after ANS Stimulation". In: *Autonomic Nervous System Monitoring*. Ed. by Theodoros Aslanidis. Rijeka: IntechOpen, 2020. Chap. 2. DOI: 10.5772/intechopen.89827.

[110] Florin Leon, Sabina-Adriana Floria, and Costin Badica. "Evaluating the effect of voting methods on ensemble-based classification". In: July 2017, pp. 1–6. DOI: 10.1109/INISTA.2017.8001122.

[111] George Barnum, Sabera Talukder, and Yisong Yue. *On the Benefits of Early Fusion in Multimodal Representation Learning*. 2020. arXiv: 2011.07191 [cs.LG].

[112] Benjamin Tardy, Jordi Inglada, and Julien Michel. "Fusion Approaches for Land Cover Map Production Using High Resolution Image Time Series without Reference Data of the Corresponding Period". In: *Remote Sensing* 9.11 (2017). ISSN: 2072-4292. DOI: 10.3390/rs9111151.

[113] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41.2 (Feb. 2019), pp. 423–443. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2798607.

[114] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. "On the Surprising Behavior of Distance Metrics in High Dimensional Space". In: *Database Theory — ICDT 2001*. Ed. by Jan Van den Bussche and Victor Vianu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 420–434. ISBN: 978-3-540-44503-6.

[115] Yang Yang et al. *TI-CNN: Convolutional Neural Networks for Fake News Detection*. 2023. arXiv: 1806.00749 [cs.CL].

[116] Yang Liu and Yi-Fang Wu. "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11268.

[117] Natali Ruchansky, Sungyong Seo, and Yan Liu. "CSI: A Hybrid Deep Model for Fake News Detection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: Association for Computing Machinery, 2017, pp. 797–806. ISBN: 9781450349185. DOI: 10.1145/3132847.3132877.

[118] Kai Shu et al. "DEFEND: Explainable Fake News Detection". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 395–405. ISBN: 9781450362016. DOI: 10.1145/3292500.3330935.

[119] Yi-Ju Lu and Cheng-Te Li. "GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 505–514. DOI: 10.18653/v1/2020.acl-main.48.

[120] Yuan Dong et al. "Performance Evaluation of Early and Late Fusion Methods for Generic Semantics Indexing". In: *Pattern Anal. Appl.* 17.1 (Feb. 2014), pp. 37–50. ISSN: 1433-7541. DOI: 10.1007/s10044-013-0336-8.

[121] Jason Brownlee. *How to Develop a Horizontal Voting Deep Learning Ensemble to Reduce Variance*. https://machinelearningmastery.com/horizontal-voting-ensemble/. Accessed on 2023-09-24. Dec. 2020.

[122] Sanaz Adel Alipour, Rita Orji, and Nur Zincir-Heywood. "Behaviour and Bot Analysis on Online Social Networks". In: *International Journal of Technology and Human Interaction* 19.1 (Aug. 2023), pp. 1–19. DOI: 10.4018/ijthi.327789.

[123] Jon Martindale. *What is Reddit? A Beginner's Guide to the Front Page of the Internet*. Sept. 2021. URL: https://www.digitaltrends.com/computing/what-is-reddit/.

[124] Peter Reuell. *Using science to combat the techniques of fake news*. 2018. URL: https://news.harvard.edu/gazette/story/2018/04/researchers-want-to-use-science-to-combat-the-techniques-of-fake-news/ (visited on 10/08/2023).

[125] Christina Pazzanese. *Social media used to spread, create COVID-19 falsehoods*. 2020. URL: https://news.harvard.edu/gazette/story/2020/05/social-media-used-to-spread-create-covid-19-falsehoods/ (visited on 10/08/2023).

[126] Semrush. *Semrush Rank - Top Websites by Traffic*. Oct. 2023. URL: https://www.semrush.com/website/top/united-states/all/.

[127] Semrush. *Semrush Rank - Top Websites by Traffic*. Oct. 2023. URL: https://www.semrush.com/website/top/global/all/.

[128] Betty van Aken et al. *Challenges for Toxic Comment Classification: An In-Depth Error Analysis*. 2018. arXiv: 1809.07572 [cs.CL].

[129] Richard Sproat and Thomas Emerson. "Normalization of non-standard words". In: *Natural Language Engineering* 9.3 (2003), pp. 289–306.

[130] Mostafa Elmasry, Taysir Soliman, and Abdel-Rahman Hedar. "Sentiment Analysis of Arabic Slang Comments on Facebook". In: *International Journal of Computers and Technology (IJCT)* 12 (Jan. 2014), pp. 3470–3478. DOI: 10.24297/ijct.v12i5.2917.

[131] Zhengqi Pei, Zhewei Sun, and Yang Xu. "Slang Detection and Identification". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 881–889. DOI: 10.18653/v1/K19-1082.

[132] Neil Houlsby et al. *Parameter-Efficient Transfer Learning for NLP*. 2019. arXiv: 1902.00751 [cs.LG].

[133] Leonard Richardson. "Beautiful soup documentation". In: *April* (2007). Accessed: 2023-07-15. URL: https://tedboy.github.io/bs4_doc.

[134] Dong Nguyen, Barbara McGillivray, and Taha Yasseri. "Emo, love and god: making sense of Urban Dictionary, a crowd-sourced online dictionary". In: *Royal Society Open Science* 5.5 (2018), p. 172320. DOI: 10.1098/rsos.172320.

[135] AbiWord community. *Enchant: The Spellchecking Framework*. https://abiword.github.io/enchant/. Accessed: March 10, 2023. 2021.

[136] coffee-and-fun. *Google Profanity Words*. https://github.com/coffee-and-fun/google-profanity-words. Accessed: March 10, 2023. 2022.

[137] George A. Miller. "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748.

[138] Karthik Raghunathan and Stefan Krawczyk. "CS 224 N : Investigating SMS Text Normalization using Statistical Machine Translation". In: 2009.

[139] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[140] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. "JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 229–234.

[141] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[142] Kai Nakamura, Sharon Levy, and William Yang Wang. *r/Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection*. 2020. arXiv: 1911.03854 [cs.CL].

[143] Alphabet's Conversation AI team. *Toxic Comment Classification Challenge*. Kaggle. Accessed: 2023-07-15. 2018. URL: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

[144] Victor Zhou. *profanity-check*. https://github.com/vzhou842/profanity-check. Accessed: March 10, 2023.

[145] Roman Inflianskas. *profanity-filter*. https://github.com/rominf/profanity-filter. Accessed: March 10, 2023.

[146] Ben Friedland. *profanity*. https://github.com/ben174/profanity. Accessed: March 10, 2023.

[147] Kishore Papineni et al. "BLEU: A Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135.

[148] *English Sarcasm Detector*. https://huggingface.co/helinivan/english-sarcasm-detector.

[149] *Sarcasm News Detection Using Roberta-Base*. https://github.com/sadia-sust/Sarcasm-News-Detection-Roberta-Base.

[150] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. "A Large Self-Annotated Corpus for Sarcasm". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[151] Zhiwei Jin et al. "Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs". In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. Mountain View, California, USA: Association for Computing Machinery, 2017, pp. 795–816. ISBN: 9781450349062. DOI: 10.1145/3123266.3123454.

[152] En Yu et al. "Adaptive Semi-Supervised Feature Selection for Cross-Modal Retrieval". In: *IEEE Transactions on Multimedia* 21.5 (2019), pp. 1276–1288. DOI: 10.1109/TMM.2018.2877127.

[153] Yi Shao et al. "Fake News Detection Based on Multi-Modal Classifier Ensemble". In: *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*. MAD '22. Newark, NJ, USA: Association for Computing Machinery, 2022, pp. 78–86. ISBN: 9781450392426. DOI: 10.1145/3512732.3533583.

[154] Dhruv Khattar et al. "MVAE: Multimodal Variational Autoencoder for Fake News Detection". In: *The World Wide Web Conference*. WWW '19. San Francisco, CA, USA: Association for Computing Machinery, 2019, pp. 2915–2921. ISBN: 9781450366748. DOI: 10.1145/3308558.3313552.

[155] Kamonashish Saha and Ziad Kobti. "DeBERTNeXT: A Multimodal Fake News Detection Framework". In: *Computational Science – ICCS 2023: 23rd International Conference, Prague, Czech Republic, July 3–5, 2023, Proceedings, Part II*. Prague, Czech Republic: Springer-Verlag, 2023, pp. 348–356. ISBN: 978-3-031-36020-6. DOI: 10.1007/978-3-031-36021-3_36.