False Textual Information Detection, A Deep Learning Approach

Fatima Taha Alkhawaldeh

PhD Thesis

University of York

Computer Science

February, 2022

ABSTRACT

Many approaches exist for analysing fact checking for fake news identification, which is the focus of this thesis. Current approaches still perform badly on a large scale due to a lack of authority, or insufficient evidence, or in certain cases reliance on a single piece of evidence.

To address the lack of evidence and the inability of models to generalise across domains, we propose a style-aware model for detecting false information and improving existing performance. We discovered that our model was effective at detecting false information when we evaluated its generalisation ability using news articles and Twitter corpora.

We then propose to improve fact checking performance by incorporating warrants. We developed a highly efficient prediction model based on the results and demonstrated that incorporating is beneficial for fact checking. Due to a lack of external warrant data, we develop a novel model for generating warrants that aid in determining the credibility of a claim. The results indicate that when a pre-trained language model is combined with a multi-agent model, high-quality, diverse warrants are generated that contribute to task performance improvement.

To resolve a biased opinion and making rational judgments, we propose a model that can generate multiple perspectives on the claim. Experiments confirm that our Perspectives Generation model allows for the generation of diverse perspectives with a higher degree of quality and diversity than any other baseline model.

Additionally, we propose to improve the model's detection capability by generating an explainable alternative factual claim assisting the reader in identifying subtle issues that result in factual errors. The examination demonstrates that it does indeed increase the veracity of the claim.

Finally, current research has focused on stance detection and fact checking separately, we propose a unified model that integrates both tasks. Classification results demonstrate that our proposed model outperforms state-of-the-art methods.

ACKNOWLEDGEMENTS

First and foremost, praises and thanks to the God, the Almighty, for his showers of blessings throughout my research work to complete the research successfully.

I would like to thank the following people, without whom I would not have been able to complete this research, and without whom I would not have made it through my PhD degree!

- I would like to express my deep sincere gratitude to my supervisors Dr. Tommy Yuan and Dimitar Kazakov for the continuous support of my PhD study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my PhD study.

- Besides my advisors, I would like to thank Dr. Peter Nightingale as a TAP member of my research degree, for his encouragement and his support during my research.

- I am extremely grateful to my parents, Taha and Imtiyaz, for their love, prayers, caring and sacrifices for educating and preparing me for my future.

- I would like to say a special thank you to my husband, Ahmad for his love, understanding, prayers and continuing support to complete this research work.

- Also, I express my thanks to all my family (my brothers and their wives and my sisters with their husbands and all their children) for their support and valuable prayers.

- I would like to express my appreciation to Al Albayt University for sponsoring my education from 2018 to 2022.

- Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

DECLARATION

I declare that this thesis is a presentation of original work, and I am the sole author. This work has not previously been presented for an award at this, or any other, university. All sources are acknowledged as references. In this thesis, some materials appeared in the published or submitted papers that are listed on page v-vi.

FATIMA TAHA ALKHAWALDEH
FEBRUARY, 2022

PUBLICATIONS

The research documented in this thesis has been published in journals and conferences as follow:

1. F. T. Al-Khawaldeh, ‘Linguistic Style-Aware Hybrid Model for Cross-Domain Factuality Checking’, *Journal of Applied Science and Computations JASC*, vol. VII, no. IV, pp. 7–20, 2020.

2. F. T. Al-Khawaldeh, ‘Hierarchical Reinforcement Learning for Factual Claim Generation’, in *Proceedings of the 1st International Conference on Automation and Artificial Intelligence. Journal of Telecommunications System & Management.*, vol. 9, no. 4, p. 23, 2020, London UK.

3. F. T. Al-Khawaldeh, ‘Factual or Non-Factual Claim : Verifying Claims’, *International Journal of Advanced Studies in Computer Science and Engineering IJASCSE*, vol. 9, no. 11, pp. 1–8, 2020.

4. F. T. Al-Khawaldeh, T. Yuan, and D. Kazakov, ‘RL-GAN Based Toulmin Argument’, *Journal of Applied Science and Computations JASC*, vol. VII, no. III, pp. 106–120, 2020.

5. F. T. Al-Khawaldeh, T. Yuan, and D. Kazakov, ‘Integrating Stance Detection and Factuality Checking’, *International Journal of Advanced Studies in Computer Science and Engineering IJASCSE*, vol. 9, no. 3, pp. 1–17, 2020.

6. F. T. Al-Khawaldeh, T. Yuan, and D. Kazakov, ‘A Novel Model for Enhancing Fact Checking’, in *Proceedings of the 2021 Computing Conference*, vol. 284, pp. 661–677, 2021.

7. F. T. Al-Khawaldeh, T. Yuan, D. Kazakov, ‘Warrant Generation Through Deep Learning’, *Computer Science & Information Technology (CS & IT)*, vol. 11, no. 20, pp. 53–75, Nov. 2021. *Proceedings of the 7th International Conference on Natural Language Computing* (NATL 2021) November 27 ~ 28, 2021, London, United Kingdom.

8. F. T. Al-Khawaldeh, T. Yuan, D. Kazakov, ‘Warrants Generations Using a Language Model and a Multi-Agent System’, *International Journal on Natural Language Computing (IJNLC)*, vol. 10, no. 6, pp. 17–26, 2021.

9. F. T. Al-Khawaldeh, T. Yuan, D. Kazakov, Perspectives Generation via Multi-Head Attention Mechanism and Common-Sense Knowledge, to be appear in the *International Conference on NLP, Data Mining and Machine Learning (NLDML 2022)*, Virtual Conference, to be included in the proceedings published by *International Journal on Cybernetics & Informatics (IJCI)*, March 12 ~ 13, 2022.

10. F. T. Al-Khawaldeh, T. Yuan, D. Kazakov, ‘Multi-Task Learning Framework for Stance Detection and Veracity Prediction’, *Semantic Web*, 2022. Under review.

Additionally, publications were made throughout the doctoral programme.

11. F. T. Al-Khawaldeh, ‘Speculation and Negation Detection for Arabic Biomedical Texts’,*World of Computer Science and Information Technology Journal*, vol. 9, no. 3, pp. 12–16, 2019.

12. F. T. Al-Khawaldeh, ‘Hierarchical Attention Generative Adversarial Networks for Biomedical Texts Uncertainty Detection’, *International Journal of Advanced Studies in Computer Science and Engineering IJASCSE*, vol. 8, no. 6, pp. 1–12, 2019.

Table of Contents

Chapter 1: Introduction

Chapter 2: Fundamental Concepts

Chapter 3: Related Works

Chapter 6: Warrant Generation Models

Chapter 7: Factual Claim Generation

Chapter 8: Perspectives Generation with Multi-Head Attention Mechanism and Common-Sense Knowledge

Chapter 9: Unifying False Information Detection Subtasks

Chapter 10: Conclusion and Future Work

List of Figures

List of Tables

Chapter One: **Introduction**

1.1. Background and Rationale

The growth of social media platforms such as Facebook and Twitter has accelerated the dissemination of false information. The study of how and why rumours and false information spread on social media have become increasingly relevant, as social media is a primary source of information dissemination. False information has the potential to influence an individual's or society's perspective on specific issues. False knowledge propagation has a number of negative consequences, most notably in the fields of politics, economics, and finance[1], [2]. We focus on fake news in this work because it is one of the domains where the greatest need exists and continues to exist as a result of the massive amount of online content that leads people to see and share information that is partially or entirely false based on social media data (e.g., Twitter) or other online sources.

There are different subtasks for false information detection to handle this problem: stance detection, fact checking, and rumour detection. First of all, stance detection [3]–[6] is the automatic categorization of the author's attitude toward a target into agreed, disagreed, discussed, or unrelated. Fact checking [7]–[9] is the task of verifying the claim, whether it is true or not. When a specific argument needs to be validated based on multiple evidential records, it may apply to the stance detection task, which is generally formulated as a multi-task classification issue. On the other hand, rumour detection is A system that identifies early-stage posts whose veracity is in doubt and alerts users that the information may be false by categorising posts as rumour or non-rumour when the information is unverified at the time of publishing. Social media posts are inputs, and a classification system is used to classify them [10], [11].

Early methods rely on feature engineering [12] that uses hand-engineered features like N-grams, part-of-speech, and sentiment features, which is time-consuming. Current research applies deep learning-based methods [7], [13], [14], which demonstrates performance improvements without the necessity to discover handcrafted features [15], [16]. Results obtained by deep learning-based methods are better than those obtained using feature engineering for solving false information problems, e.g., Recurrent Neural Networks (RNNs) for representing sequential posts and user engagements for twitter rumours [10], [11], [17], [18], Convolutional Neural Networks (CNNs) for capturing local features of texts and images [19], and Generative Adversarial Networks (GANs) [20] for capturing deceptive writing style features. In general, enormous datasets are required for deep learning models to capture additional characteristics and features, and deep learning models lack a clear interpretation of the result due to their black-box nature and undue complexity.

Despite the recent advances with false information detection technology and the application of deep neural networks, several practical issues have yet to be fully addressed, and there are still some challenges, as illustrated in figure 1.1.

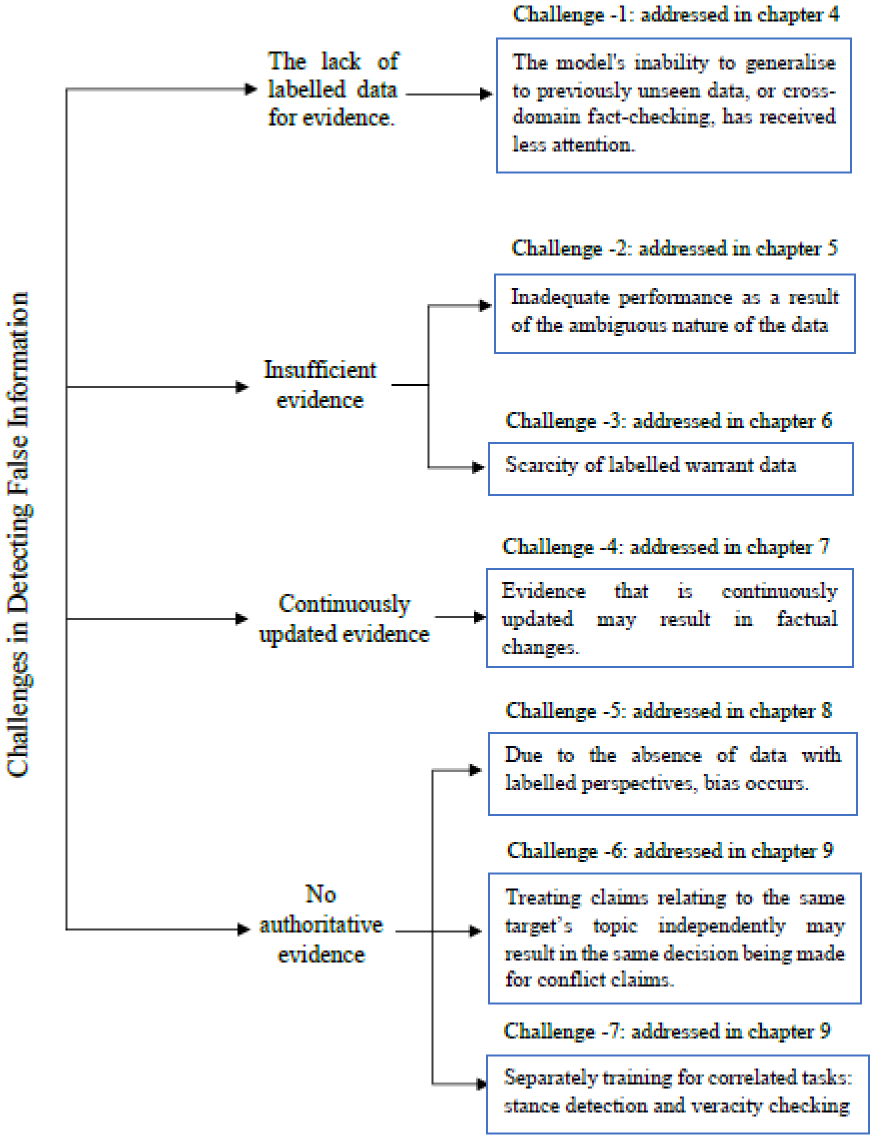

Challenges in Detecting False Information
The lack of labelled data for evidence.
Challenge -1: addressed in chapter 4
The model's inability to generalise to previously unseen data, or cross-domain fact-checking, has received less attention.
Insufficient evidence
Challenge -2: addressed in chapter 5
Inadequate performance as a result of the ambiguous nature of the data
Challenge -3: addressed in chapter 6
Scarcity of labelled warrant data
Continuously updated evidence
Challenge -4: addressed in chapter 7
Evidence that is continuously updated may result in factual changes.
No authoritative evidence
Challenge -5: addressed in chapter 8
Due to the absence of data with labelled perspectives, bias occurs.
Challenge -6: addressed in chapter 9
Treating claims relating to the same target's topic independently may result in the same decision being made for conflict claims.
Challenge -7: addressed in chapter 9
Separately training for correlated tasks: stance detection and veracity checking

Figure 1.1: An overview of challenges in detecting false information

Firstly, there has been little attention paid to limited labelled evidential data and the lack of evidence for emerging claims, which makes it difficult to verify claims using knowledge bases, especially new claims. Every second, a vast amount of data is created and made available on the web in multiple domains in this era of constantly spread data. As newly emerging claims checking is the first step toward mitigating rumours' negative effects, such as false beliefs and bias public opinion regarding social and political decisions, it is necessary to address this challenge by enhancing the generalisation performance of models that address the scarcity of evidence for emerging claims (unseen posts). Model generalisation is the process of transferring a model trained on a source domain with a large amount of labelled data to a target domain with a scarcity of labelled data. The disadvantages of prior methods include their inability to be applied to new data sets as they train their models on domain specific information as in Gravanis et al. [21]. This complicates the process of adapting their methods to a new claim (emerging rumours). As a result, a novel methodology for verifying the veracity of claims against unseen authoritative sources is required, improving the generalisation performance and transferability of false information detection models to new machine learning models' data. Domain generalisation's goal is to combine knowledge from multiple source domains into a single model that can generalise well to unseen target domains where a large number of claims are made quickly and without evidence at the time of posting. Chapter 4 proposes a novel model to address this issue.

Secondly, a lack of sufficient data will adversely impact the overall performance of fact checking models. One of the major challenges for fact checking is to understand the correct label of a claim against the knowledge base (or evidence) and how the conclusion is reached. Even though the relevant evidence has a significant role in deciding the claim's factuality, if the relationship between the claim and its relevant evidence could not be detected correctly, the wrong decision of factuality could result from insufficient evidence. The majority of current systems depend entirely on evidence to determine the relationship between the argument and the evidence, resulting in incorrect fact labels. For instance, works in which FEVER has been used to conduct experiments[8] and which rely solely on available evidence. In other words, when a model is unable to explain the evidence clearly, it is less likely to perform well. Therefore, it is necessary to resolve insufficient evidence by acquiring additional clarification (e.g., warrant) to assess the claim's credibility where warrant is a logical inference statement that acts as a link between the claim and the evidence to extract supportive sentences. The solution to this problem is proposed in chapter 5.

Thirdly, related to the second challenge, the shortage of labelled data for additional warrants poses a challenge to the performance of models. Despite the importance of additional information (warrant) for comprehending and correctly linking a claim to evidence, the absence of labelled warrant data creates a problem for fact checking in determining the claim's factuality. Manually annotating data and managing labels to extract or locate this data is a time-consuming process. As a result, it is essential to resolve the shortage of labelled data to maximise the benefits of deep learning. It would be helpful to prompt the development of a model capable of generating this type of information to overcome data scarcity. An algorithm can then check the claim's veracity without the need for external evidence, and this is expected to address the scarcity of the annotated data and the continuity of publishing new and unseen information. This challenge is discussed in chapter 6.

The fourth difficulty is related to the possibility that evidence will change over time, necessitating the establishment of a mechanism for correcting written claims that are refuted or only partially supported by evidence. Fact checking entails classifying assertions as true or false without rewriting them to be more consistent with the retrieved evidence. Fact checking models face a challenge due to the scarcity of datasets containing claims and their corrections. This creates an issue: the reader's inability to discern subtle issues that result in factual inaccuracies. A novel model for the generation of factual claims with re-purposed data, is proposed to address this issue and provide an explanation for the faculty claim decision. Chapter 7 examines a framework for resolving this problem.

The fifth challenge is the shortage of data from alternative perspectives (viewpoints), which often contributes to biased labelling. Rather than analysing an argument from a single viewpoint, it should be analysed from a variety of angles and their respective attitudes toward it. The internet contains several sources of content, including news websites, blog posts, social media platforms, and message boards. Despite the abundance of useful knowledge available from various sources, manually extracting perspectives is extremely difficult. Existing methods for verifying social media posts were evaluated over existing data (i.e., the claim's information is easily accessible).

Due to the difficulty of identifying replies related to the claim and other potential alternatives, the current approaches are biased due to the lack of consideration of alternative viewpoints. As a result, verifying statements made using various data sources is restricted compared to data generated during real-world related replies. This complexity is magnified for fact checking models and deep learning models since they require large amounts of manually labelled training data. To fix the bias problem, where the only evidence is considered, and to maximise the benefits of deep neural network algorithms, it is critical to overcoming data scarcity from other perspectives. There is a need to generate a diverse range of perspectives from trustworthy sources to evaluate the veracity of a claim to eliminate the biases and the scarcity of relevant perspectives. The generated claim must draw a more accurate inference from the evidence, which is unambiguous and correctly summarises the evidence. No early works have used generation tasks to produce factual claims that could interpret the rationale for a claim's veracity. Chapter 8 examines a strategy for resolving the situation.

A further challenge, challenge 6, is that most existing approaches for false information detection are dedicated to an individual task, i.e., fact checking and stance detection as separate models, rather than a combination of them. For example, given a claim with several documents and each document with a particular stance toward the claim, the stance detection task is an earlier phase for claim factuality detection. With separate training for the sub-tasks of false information detection, it is challenging to treat them separately while highly correlated. A model that considers different stances to detect a claim's veracity is required to address this. Chapter 9 discusses a possible model for resolving this issue.

The last challenge, challenge 7, the existing literature does not conceptualise bipolar argumentation and truth discovery as a single construct containing contradictory information. Bipolar argumentation is an extension of Dung's argumentation framework [22]. A bipolar argumentation encompasses both attacks and supports between arguments [22]. Despite the highly related connection between truth discovery and bipolar argumentation, none of the literature works reframes truth discovery to investigate argumentation-based Truth Discovery. The ninth chapter looks at a strategy for dealing with the situation. We propose argument-based truth discovery as a possible solution for combining stance detection and claim veracity detection. Thus, investigating each claim independently concerning the same target's topic may result in the same label for contradictory claims.

By addressing these limitations, this thesis contributes to the field of false information, with a focus on fake news. The following section discusses the research questions that need to be addressed to produce an automated false information detection framework.

1.2. Research Questions

To address the challenging research issues discussed in the previous section, we formulate many research questions and hypotheses.

RQ 1. Can we improve the state of art performance for emerging claim verification?

H1. For new/emerging claims, the ability to extend the existing framework to new contexts or handle new data needs to be demonstrated. We hypothesise that the writing style (the linguistic properties of claim) could provide richer information and guide the fact checking tasks to classify the type of information as false or true, without the relevant evidence. We hypothesise that subjective and biased languages generate more false claims and vice versa and that detecting the writing style improves the model's performance.

RQ 2. To what extent can external knowledge, such as a warrant, aid in fact checking performance improvement?

H2. To address this research question, we hypothesise that if a warrant could connect the evidence and the claim, i.e., provide the rationale for supporting the claim, they can be used to improve false information detection performance.

RQ 3. How to generate high quality and more diversity warrants?

H3. To answer the research question, we believe that contextual information can be incorporated into deep learning through the use of natural language processing techniques such as Rhetorical Structure Theory, which is a language-independent descriptive theory of text organisation and discussed in detail in section 2.3.6, and causality for improved warrant generation. Also, it is possible to improve the diversity and quality of warrants by fine-tuning a pre-trained Language Model (BART) using Multi-Agent Network reinforcement learning.

RQ 4. To what extent does the generation of factual claims, as explanation for the reason behind the decision, affect false information detection performance?

H4. We use factual claim generation to aid in the detection of false information. Rather than verifying the claim using evidence that may contain a great deal of information about a variety of different topics or targets, we hypothesise that a factually generated claim contains more concise information about the claim to be verified.

RQ 5. Does incorporating common-sense knowledge improve the performance of deep learning to generate several relevant candidate perspectives for a given claim?

H5. To avoid biases, we try to produce many perspectives that could be used later to retrieve relevant evidence from search engines to predict the claim's credibility. Thus, if the stance detection is performed for all claim-perspectives pairs, and then the decision of factuality is taken, the issue of unlabelled perspectives will be addressed. We propose that encoding claim-aware common-sense during perspective generation improves the diversity and quality of generated warrants and outperforms current models.

RQ 6. To what extent would combine the sub-tasks of false information detection, such as stance detection and veracity checking, improve its performance?

H6. Combining fact detection with stance detection tasks can improve the performance of claim credibility detection.

RQ 7. To what extent can the bipolar argumentation framework be a potential solution for multitask truth discovery problems and improve false information detection performance for conflict claims?

H7. For conflict sources with uncertain authenticity, mapping the truth discovery network to the bipolar argumentation framework helps find the believable fact amongst conflict information.

RQ1 is addressed in chapter 4 of this thesis, whereas RQ2 is discussed in chapter 5. RQ3 is presented in chapter 6, RQ4 is covered in chapter 7, RQ5 is addressed in chapter 8, and chapter 9 concerns RQ6 and RQ7.

1.3. Thesis Overview

The rest of the thesis is organised as follows. Chapter 2 discusses concepts of false information and deep neural networks. Chapter 3 reviews related works for false information detection. Chapter 4 investigates the importance of linguistics features for fact checking to generalize to unseen data. Chapter 5 focuses on leveraging warrant and fact information to solve the poor performance problem in fact checking. Chapter 6 propose models to generate argument components (warrant). Chapter 7 discusses the factual claim generation. The means to overcome the bias problem, i.e., by generating perspectives, are discussed in chapter 8. Chapter 9 explores the means to integrate the sub-tasks for false information detection. The conclusion and future works are discussed in chapter 10.

Chapter Two: **Fundamental Concepts**

This chapter describes the background of false information and its detection. Initially, we present various definitions of false information offered by scholars in different domains, such as computer science, to understand this type of information better. Also, we discuss some characteristics of false information and circulating on the web and social media. This chapter also discusses false information detection, natural language processing for false information and detection, and deep neural network for false information detection

2.1. False Information

While fact checking was previously carried out manually by journalists, the internet has become a source of an increasing number of contentious statements from politicians, biased news reports, rumours, and others. The requirement for developing an automated model for determining the veracity of claims has increased. Since social media has risen to prominence as a news source, it has facilitated rumours and misinformation. The advancement of natural language processing and information retrieval technologies and the availability of datasets enable the automation of fact checking.

False information is classified [23] into two primary categories based on the publisher's intent: misinformation and disinformation. Misinformation is the unintentional dissemination of false information, for example, misinterpretation of factual information and subsequent delivery of it with different facts [24]–[27]. Disinformation is the dissemination of false information with the intent of deceiving the reader, such as advertisements, campaigns, and influencing individuals' or society's public attitudes and beliefs [28]–[30]. According to its content, published material can be classed as either opinion-based or fact-based. While opinionated content reflects the subjective perspectives of individuals, faked reviews are considered false information. In the news domain, when a fact is fabricated or contradicts previously trusted information, this is known as fake news, and it is the primary target of our work. It is categorised in one of the following ways:

- Propaganda is the practice of deceiving the public for political purposes [31].
- News fabrication is true statements are altered in such a way that their meaning and intent are altered, convincing people that they are legitimate [32], [33].
- Imposter content is when authentic sources are impersonated by fabricated ones.

- Satire or parody which has no malicious purpose yet has the capacity to deceive. The primary distinction between satire and parody is that satire employs humour, irony, exaggeration, or ridicule to expose and criticise people's stupidity or vices, whereas parody employs deliberate exaggeration for comic effect.
- Manipulated content is evidence that has been altered.
- Advertising is a strategy for bringing products or services to the public's attention, primarily through paid advertisements. Public relations is a form of strategic communication that focuses on the establishment of mutually beneficial relationships between businesses and their customers.
- The term "false content of connection" refers to irrelevant content or a title that is unrelated.
- The term "conspiracy theory" refers to a situation designed to arouse public outrage for political purposes [34].
- A hoax is the spread of false information under the guise of fact.[35].
- Biased information that is overwhelmingly slanted in one direction [36], [37].
- Rumours are unconfirmed information at the time of posting [38]
- Clickbait is content created with the primary goal of capturing visitors' attention and convincing them to click on a link to a specific web page [39].

2.1.1. Who Spreads False Information and Why?

One reason for spreading false information is to be the first to publish, regardless of the information's validity, to increase the publisher's views. Another reason could be to enable a large audience to receive the same information quickly to accomplish political objectives [30], [40], [41]. Others attempt to increase social media interaction to disseminate false information by sharing or interacting via liking, disliking, or commenting, thereby increasing the likelihood of seeing more users [42]–[44].

False information is difficult to detect, and it spreads rapidly due to its credible and legitimate appearance, and people are poor judges of false information [45], such as hoaxes and fake reviews. Even if the motivation for disseminating false information is benign and not deceptive, it is still harmful because the reader receives it as factual information. Individuals with a higher level of education, experience and those who work in the media are more likely to be good judges of false information [46].

Numerous studies have been conducted to determine the impact of false information, including Facebook posts [47], [48], and articles on Wikipedia [35]. The primary issue with spreading false information is that it initially appeared to be true [49]. This information has a detrimental effect on a variety of areas, including terrorist activity [50], [51], obstructing response to natural disasters [52], and the stock market [53]. Additionally, it has been detrimental to marketing [42], entertainment [44], and the excitement surrounding an idea, individual, or organisation [54], as well as increased advertising revenue for websites [55], as well as advancing a particular entity's agenda [56].

According to Zannettou et al. [57], several actors are critical in the propagation of fake news:

- Bots [58]: automated applications that generate false information based on fake accounts.
- Criminal/Terrorist Organizations [59]: terrorist and extremist groups use social media to spread false information resulting in terrorist actions.
- Activist or political organizations [60]: political parties which disseminate false and untrue information, promoting their organisation or downgrading other competitors.
- Government [61] shares fake news to change public opinion on specific political issues.
- Hidden paid posters and state-sponsors trolls [62], [63]: a group of people post false information that serves an agenda to influence public opinion of social or business matters
- Tendencies. This type of actor has the same profit purposes as bots.
- Journalist [64]: people share false information to either the online or offline world to increase their popularity.
- Useful idiots [65]: users who fabricate false information to strengthen the effectiveness of the organization's marketing campaign.
- True believers and conspiracy theorists [66]: when people share false information unconsciously.
- The individual who receives false information. Users create fake information for commercial and/or personal benefit.
- Trolls [67]aim to do things to annoy or disturb other users and share false information to provoke or annoy other users.

2.1.2. False Information Characteristics

Various researches have been conducted to examine the characteristics of false information, which can be of assistance in determining the legitimacy of the propagated information, including fake reviews [68], fake news [69], and hoaxes [35]. Numerous studies describe the characteristics of false information to differentiate it from genuine information, e.g., textual content, temporal features, ratings, references, and user properties.

As a guide for determining its veracity, a large body of literature analyses the textual characteristics of false information [48], [49], [70]. Rubin et al. [71] examined satire and discovered that it contains fewer words than real news and fewer technical and analytical words, while others examined fake news and discovered that it contains fewer nouns and more verbs, particularly on social media [69]. Others demonstrate the use of more ambiguous and hedge words to express false information [72].

Apart from textual body characteristics, some studies, such as Kumar et al. [35], concentrated on the characteristics of the false information creator. According to their analysis, hoaxes created by registered accounts contain a high volume of text but few references. Automated accounts that quickly spread the rumour are primarily bots [30], [73].

2.2. False Information Detection

The process of fact checking involves comparing a claim to relevant evidence and determining its truthfulness. Stance detection identifies each document's stance concerning the claim and then predicts the claim's factuality by aggregating the strength of the stances while considering the source's credibility[74]. It is often argued that objective articles are more likely to generate valid claims than subjective articles (or articles that contain false information) [9], [75], [76]. For instance, Mukherjee & Weikum [77] argue that in an objective article, if a claim's objectivity score (based on facts and observations) is greater than its subjective score (influenced by personal feelings), the claim is more likely to be true, and vice versa.

Argument mining techniques aid in the comprehension of the relationships between claims and other relevant data, such as evidence [78]. In the ever-changing environment of social media, the presence of argumentative features can aid scholars in their efforts to combat rumour spreading, identify fake news, and cite sources. Automatic argument evaluation has the potential to reduce the spread of rumours, accelerate the identification of fake news, and ultimately improve the quality of public political discourse [79].

Potthast et al. [36] classified the literature on fake news detection into three categories: content-based, context-based, and style-based, overall, content-based mechanisms are the only ones that are likely to result in objective determination, whereas context- and style-based mechanisms are likely to be less reliable.. Certain approaches combine content-based and context-based methods; for example, GAN-based detection models [80], RNN [17], hierarchical neural networks [81], and graph neural networks [82], [83] consider the information contained in text bodies and user profiles when detecting false information. This section reviews existing methods for detecting false information, emphasising the three types of methods.

2.2.1. Content-based Methods

The simplest way of identifying false information is to examine the veracity of claims made in news and information. Content-based approaches are frequently referred to as fact checking. It presents criteria for judging the truth of a claim that validates one piece of evidence. For instance, a claim is false if the content has contradicted fact, or the content claim is worded in such a way that it omits to disclose the true content indicated in the evidence article.

There are several concerns for content-based fact checking, like depending heavily on expressive features such as Term Frequency and Inverse Document Frequency (TF-IDF) features [84]. Karadzhov et al. [85], depending on ground truth as a credible source like google engine, applied LSTMs set on retrieved results to enrich SVMs and multilayer perceptron to detect the factuality. Evidence is extracted by searching from trusted websites to verify news based on the claim queries method. A dataset consisting of 992 sets of tweets is used for experiments. Despite satisfactory results by following question answering, which needs generating queries and selects the best snippet than the best sentences, it is incredibly important to get the final factuality label. The system has the benefit of not depending on highly engineered features. In this automatic system, the information they gain comes from the Web, the accuracy may be affected according to how much the retrieved snippet is relevant and to what degree the sources are trusted.

Recently, there is much Content-based approaches concerning false information checking; for example, 63% of accuracy has been achieved by applying the discourse feature similarity in Rubin et al. [32]. Support Vector Machine (SVM) shows promising results when implemented on the BBC, the 20 Newsgroup and other news datasets [86], [71], obtaining 94.93 %, 97.84 % and 90 % accuracy, respectively. Conflicting views are studied, where the accuracy reached 84 % [87]. Harmonic Boolean label crowdsourcing algorithm is applied with 99 % accuracy [88]. Ahmed et al.'s work [89] employs N-gram analysis with ML methods, reaching 90% accuracy. Attention-based long-short memory network is used in Long et al. [90], outperforms Yang et al.'s work [91] where the convolutional neural network is applied and evaluated Fake news dataset benchmark FND, with 41.5 % accuracy. Guacho et al. [92] applied the content-based FND method with 72 %, 61.3 %, and 70 % accuracy for three datasets, two datasets provided by [70] and [93], and the third is their fake new dataset. In Ozbay & Alatas [94], the adaption of the two metaheuristic algorithms, the Grey Wolf Optimization (GWO) and Salp Swarm Optimization (SSO) show promising results compared to the state-of-the-art.

2.2.2. Context-based Methods

In contrast to content-based methods, which consider a claim's veracity regardless of the other posts or responses to it. Context-based methods take into account additional characteristics, such as those described by Hanselowski et al. [95], user profiles and their credibility [96]–[98], and the relationship between a claim and the stories that interact with it (e.g., commenting) [69][99][100] [49].

Network-based methods are also referred to as context-based methods, depending on their application, which can be static or dynamic. Static networks of users (created or derived from) on social media in terms of interests, topics, relationships, and dynamic networks created through the propagation of false information provide numerous credibility features. Methods based on networks [43], [55], [97], [98], [101] analyse networks in general to determine the transmission subtree, depth, degree distribution, and clustering coefficient. Existing network metrics are used to determine the difference between true and false information; for example, Jin et al. [43] use conflicting social viewpoints in a credibility propagation network to automatically verify microblog news.

Recurrent/Recursive Neural Networks (RNNs) are used to represent sequential posts and user engagements [10], [11], [17], [102] and tweet propagation data. In Ma et al.'s [10] system, fake news propagation is represented by RNNs composed of bottom-up and top-down tree-structured neural networks trained on user properties and profiles. Ruchansky et al. [17] developed a Capture-Scoring-Integrated (CSI) model consisting of three components: The Capture module uses LSTM to extract textual information about a user's pattern of temporal engagement with an article. While the Scoring module detects all users' source characteristics, the combination of article representation from the first module and user information from the second module occurs in an integrated module. The relationship between news creators and subjects is deduced using an RNN model [102]. The LSTM algorithm is used to extract temporal textual characteristics (time-series events) from Twitter rumours. Even this system can learn without extensive manual training; hidden representations of these networks take a long time to detect dynamic structures [11].

Context models, particularly stance-based models, attempt to represent users' social responsibility in terms of stance to predict how the news will be perceived [87]. At the same time, content-based approaches make use of linguistic characteristics to alert users to false information [36].

2.2.3. Style-based Text Classification

Generally, in style-based methods, the propagated information is delivered via a short text, story, or article, and the writing styles, such as lexical, syntactic, and topical features, are the primary features used to detect false information [103]–[105], for example, emotional features, keywords, and message tenses [103]. Castillo et al. [103], [104] demonstrate that longer textual content is more likely to be correct than shorter textual content. Kwon et al. [106] discovered that veracity checking requires cognitive and action words, negating words, and sentiment analysis. Pérez-Rosas et al. [69] noted additional distinguishing characteristics such as readability, N-grams, punctuation, and syntax, as well as psycholinguistic characteristics, such as a part of speech (POS) [107], first/second person pronouns [1], and topic and user topic features [108]. Prior research has established that n-grams are an effective feature for detecting fake news, and the majority of prior work on fake news detection has concentrated on POS features. Additionally, when readability features are incorporated into our proposed model for fake news detection, additional improvements are achieved.

Fake news articles are deceptive texts depend on the written style [109]. Rubin et al. classified fake news into three categories that are more serious than others [110]: Serious Fabrications, Hoaxes, and Satire. Conroy et al. [111] emphasised the importance of linguistic information in detecting false information. Numerous techniques are used to detect fake news. Traditionally, machine learning algorithms use a predefined set of linguistic features and a large amount of labelled data. Others, such as Horne & Adali [70] and Yang et al. [112], demonstrate significant influence by using modern neural network models based on pre-trained word vectors and embedded representations. The style of writing incorporates lexical, syntactic, and structural analysis of various levels of language construction. Style-based text classification is proposed by Argamon-Engelson et al. [113]. It is based on the count of function words and part-of-speech trigrams.

Numerous methods for classifying texts based on their stylistic features have been proposed in the literature. For example, Popat et al. [114] concentrated on manually extracted stylistic features such as the frequency of assertive and factive verbs, hedges, implicative words, report verbs, and discourse markers. While others, such as Barrón-Cedeo et al. [115], focused on style markers such as character sequences and readability measures.

Potthast et al. [36] used stylometric analysis to predict factuality (fake vs real), which was implemented for the authorship verification task proposed by Koppel et al. [116], taking into account specific characteristics such as character n-grams, part-of-speech, readability scores, and the presence of specific words. Horne and Adali's [70] study examines the writing style to determine the truthfulness of the information, taking into account readability measures for complexity scoring and specific characteristics such as negation, the frequency of occurrences of various part-of-speech tags, swearing, and slang words. They demonstrate that for authentic information, complexity is greater, and texts are longer. Conroy et al. [32] developed a style-based model for detecting fake news that employs rhetorical structure theory as a criterion for factuality. The surface-level linguistic pattern and the hybrid convolutional neural network for integrating metadata with text perform admirably [117].

The majority of deep neural networks used for style-based text classification are lexical-based language models, which are less scalable when dealing with heterogeneous data, such as multiple subjects and genres. Jafariakinabad et al. in [118] demonstrate that syntactic models are more robust to various topics than lexical-based models. They train six classifiers to determine the significance of a term and 53 other linguistic characteristics using TF-IDF cosine similarity. Gravanis et al. [21] demonstrate that combining word embedding and linguistic features improves the performance of the fake news classification system. Martino et al. [119], the shared task of fine-grained propaganda detection at the NLP4IF workshop, propose a new task in which each sentence fragment contains one of eighteen propaganda techniques such as repetition and exaggeration.

Numerous linguistic techniques exist for detecting fake news and identifying indicators of deception [111]. Feng & Hirst [120] employ n-gram and deep syntax features as indicators of deception, a hybrid approach that incorporates a convolutional neural network [117]. Popat et al. [114] consider stylistic features such as assertive and fictitious verbs. Pérez-Rosas et al. and Pierri & Ceri [69], [121] discuss the most promising approaches, taking into account various linguistic features and weighting them differently for classification.

Rashkin et al. [1] distinguish between real, satire, hoax, and propaganda using TF-IDF-weighted words; some words appear more frequently in fake news. To classify social media posts retweeted from news accounts as verified, hoax, satire, propaganda, or clickbait, linguistically infused neural network models (Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) and word embedding) are used [122]. Specific information such as tweet text and linguistic bias, subjectivity, and psycholinguistic markers is incorporated into the models. The authors demonstrate in this study that the applied model outperforms logistic regression by a significant margin.

The style of the written document is an excellent indicator of the claim's veracity. True claims are written objectively and unbiasedly, whereas false claims are written in a subjective and biased manner [75]. Numerous proposed linguistic features are examined in [123]–[125].

Earlier research has relied on various linguistic characteristics; we will use those listed by Popat et al. [126]: a set of assertive and factual verbs, subjective and biased words, vocabulary richness, and readability [115]. Sentiment, structure, bias , and complexity are all NELA characteristics [127]. Lexicon sources and lexicons for feature extraction include Wiktionary, the Linguistic Inquiry and Word Count (LIWC) lexicon [128], Wilson et al. [129], Hyland [130] and Hooper [131].

2.3. Natural Language Processing for False Information Detection

Thorne and Vlachos [132] discuss NLP related tasks for fact checking, such as researching fake news, and textual entailment, emphasising the importance of evidence in fact checking techniques. Snopes (snopes.com) and PolitiFact (politifact.com) are two examples of fact checking websites used to identify false information using evidence. Certain works incorporate an NLP component for stance detection; for example, Saikh et al. [133] combined textual entailment with stance classification using statistical machine learning and deep learning approaches.

In NLP applications, text representation is the main task for text classification, i.e., labels to the represented text. Also, there are several NLP tasks strongly related to false information detection. We first introduce deep learning components, such as word and character embedding, sequence encoder, and the attention mechanism, that form the modern basis to process textual information. Then we describe linguistic features that are widely used in the natural language processing community.

2.3.1. Natural Language Inference

Textual entailment is a directional relation between text fragments and their approaches require textual evidence for fact checking, meaning this technique can only work when text evidence is provided. To judge which claim is true and not, textual entailment models recognise a relationship between a claim and relevant evidence. Thus, textual entailment is critical in determining whether an evidence text implies a hypothesis. The effectiveness of rule-based generators was demonstrated by including noise data and adversarial loss during optimization [134] on Standard Natural Language Inference (SNLI) training [135]. To develop a rich representation of statement pairs and to detect their relationship, a machine-learning technique called an "ESIM" was used [14], [136]. The majority of inferences requiring lexical knowledge are made using MultiNI (Multi-Genre Natural Language Inference) [137]. We use the entailment metric as a rewarder in chapter 5 to select the best warrant and for perspective generation in chapter 8.

2.3.2. Word Embeddings

Text representation is critical for a large number of real-world natural language applications to function properly. For instance, the traditional one-hot encoding of words obtains no syntactic or semantic information, and models are unable to utilise information about word relationships. One hot encoding is a critical component of feature engineering for machine learning because it allows for the conversion of categorical data variables into new categorical columns and assigning those columns a binary value of 1 or 0. Each integer value is represented as a binary vector that can be used to increase the accuracy of prediction. Word embeddings address this issue by clustering words with similar meanings in the representation space, significantly improving performance.

It has been discovered that representing words as vectors is beneficial for Natural Language Processing (NLP) tasks, because they are visually appealing, have better syntactic and semantic word relationships, can be subjected to use operations such as addition and distance measures, and are well-suited for use in a variety of Machine Learning (ML) algorithms and strategies. Word embeddings are dense, distributed, fixed-length word vectors representations for words representations in a low - dimensional vector space. They are critical for language models because they can be used directly to neural network language models. Its main goal is to predict the next word based on a set of previous words.

To obtain such representations, various approaches are used to train word embeddings [138], [139]. The first approach, Prediction-based Models, use SoftMax regression to fit bigram probabilities and is optimised using Stochastic Gradient Descent (SGD) as in word2vec. The second approach, Count-based Models, use the product of two low-rank factor matrices to reconstruct certain bigram statistics matrix extracted from a corpus as the well-known GloVe.

Word embeddings are a technique for accurately representing words by encapsulating their semantic content, real-valued word representations trained on natural language corpora capable of capturing lexical semantics[140]. Language representation is a critical component of automatic models because it enables efficient methods by extracting the most valuable information from raw texts and transferring some task-specific features, such as in cross-lingual tasks [11], [141]. Various models are used to improve word representation, including reinforcement learning to select useful words [142]. By combining word embeddings and linguistic features, performance in detecting fake news is improved [21]. This section will discuss several recent methods for creating vector representations via word embedding.

Word embedding is more effective at determining the semantic meaning of words, as demonstrated by Word2Vec and Glove, which utilise both local and global context to determine the contextual information for statements. Elmo and Bert [143] are two recent powerful pre-trained deep contextual deep word representation models on a large text corpus. We represent each word in chapter 4 by using multiple word embeddings. Given a word w, we obtain the word vector using a variety of pre-trained word embeddings such as Word2vec, glove, Elmo, and FastText.

2.3.3. Word2vec

It is a powerful tool for extracting word representations from corpora that use two models: CBOW and Skip-gram (SG) [144]. Both models are capable of capturing a word's semantic information:

CBOW: The purpose of this model is to discover useful word representations for predicting the target word from the context words; as illustrated in figure 2.1, the input is the context words of the target word wt. Given context words, the sequence of words preceding and following target words based on the window size of the target word w, and utilising the similarity and SoftMax functions as shown in equation 2.1 where w′ is a word from the vocabulary V and sim (wt, wt+j) indicates the degree of similarity between the current word wt and one of its context words wt+j.

Equation 2.1 $$p\left(w_t \middle| w_{t+j}\right) = \frac{exp\left(sim\left(w_t, w_{t+j}\right)\right)}{\sum_{w' \in v} exp\left(sim\left(w', w_{t+j}\right)\right)}$$

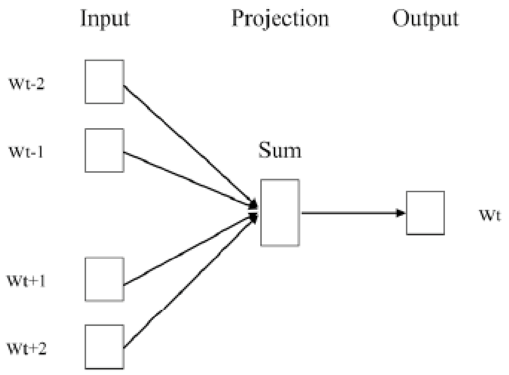


Figure 2.1: The overall architecture of CBOW when the window size is 2.

Skip-gram: The Skip-gram model identifies useful word representations for predicting the target word's context words. The model's input is the target word. As illustrated in figure 2.2, the target word is first mapped to a hidden layer vector representation, and then predicted context words are generated based on this. Similar to the CBOW model, it maximises the average log probability by utilising similarity and soft functions, as in equation 2.2 [145].

Equation 2.2 $$p(w_{t+j}|w_t) = \frac{exp(sim(w_{t+j},w_t))}{\sum_{w' \in v} exp(sim(w',w_t))}$$

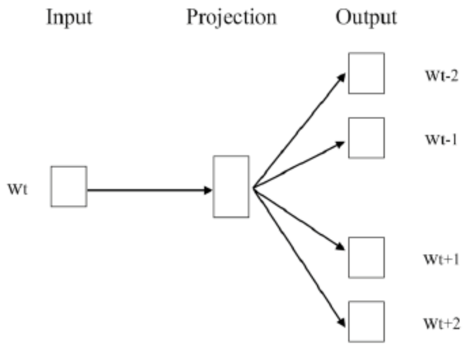


Figure 2.2: The overall architecture of skip-gram, windows size is 2.

2.3.4. Glove

Word2vec's trained word embeddings capture both the semantic characteristics of words and their overall relatedness. However, Pennington et al. [146] point out that the Word2vec model only uses information from the local context window and ignores global context-independent information, whereas the Glove model considers the global co-occurrence matrix [146, p. 75]. Thus, local context reflects a word's local semantic and syntactic characteristics; global context encodes the document's overall semantic and topical properties.

2.3.5. Elmo

This model, called Embeddings from Language Models, is used to learn how to represent text effectively to provide rich, context-dependent, and character-based lexicon representations, such as [147] Bidirectional LSTM-based language modelling. This model trains a bidirectional LSTM model on a large corpus using Elmo and then uses LSTM to generate the representations for the words. The forward LM calculates the probability of an input sequence and updates it based on historical observations for each token (i.e., previous tokens in the sequence). A backward LM determines the likelihood of each future token given the current token. Elmo employs a linear combination of the states of two bidirectional LSTM layers and the word embeddings in the input, as illustrated in figure 2.3. To improve performance, the Elmo vector can be added to the hidden state of each task-specific model. These uses should change in light of two factors: the degree of complexity in word meaning and how grammar works and the changing language environment.

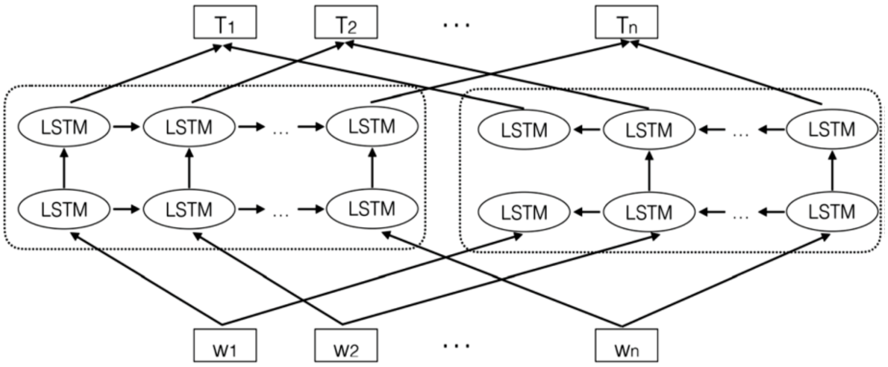


Figure 2.3: The overall architecture of Elmo

2.3.6. Relation Extraction and Rhetorical Structure Theory (RST)

RST is a framework for analysing a text's coherence. By specifying the semantic role, for example, a sentence for Evidence, this framework can identify the central idea and analyse the characteristics of the input text systematically. Then, based on its coherence and structure, it is determined whether it is fake news [148]. Additionally, researchers identified instances of deception using discourse analysis, rhetorical markers, and linguistics [149]. RST contributes to the generation of warrant in our work and will be discussed in chapter 6.

Rhetorical Structure Theory (RST) is a language-independent descriptive theory of text organisation in computational linguistics that explains text structure in terms of the relationships between the speech or rhetorical elements present within a text [149]. It creates a framework for portraying texts and their rhetorical links. A tree must interpret the text's overall structure, referred to as the RS-tree; in Rhetorical Structure Theory, the term "schemas" refers to the communicative functions of a text structure (RST). Using conceptual frameworks to represent the communicative roles of a text structure undoubtedly aids in the representation of discourse for argument mining. It describes twenty-three possible connections between textual spans. The distinction made by RST between the section of a text that serves as the author's primary objective (nucleus) and the section that serves as supplementary material (satellite) is critical when analysing argumentative texts (or schema) [150], [151].

Rhetorical Structure Theory (RST), sometimes combined with the Vector Space Model (VSM), is also used for fake news detection [32], [152] by defining the semantic role (rhetorical relations) for the coherence of a text.

2.3.7. Lexical Chain (LC)

LC is s a written collection of related terms that span short (adjacent words or sentences) or long distances a sequence of words that captures a portion of the text's coherent structure by capturing semantic similarities between noun phrases; lexical chains are used to evaluate the significance of sentences [153]. Al-Khawaldeh & Samawi [154] used LCs to group related items into chains and then separate solid chains based on scoring criteria, with the chain sequence derived from a word-net and a similarity index threshold.

The most semantically related terms are lexical chains: synonym, holonym, and meronym. The Wordnet Hierarchy and a hierarchical tree-like structure extract common senses for each term. Therefore, the warrant should be inextricably linked to the most robust chain of evidence available in our investigation. Examples of lexical chains are provided by the metric used to calculate the chain's score, equations 2.3 and 2.4:
LC1= policy, rule, strategy, procedure
LC2= features, attributes, characteristics, structures

Equation 2.3 Score (Chain)= Length of chain× Homogeneity of chain

Equation 2.4 Homogeneity=1-DistinctMembers/Length of chain

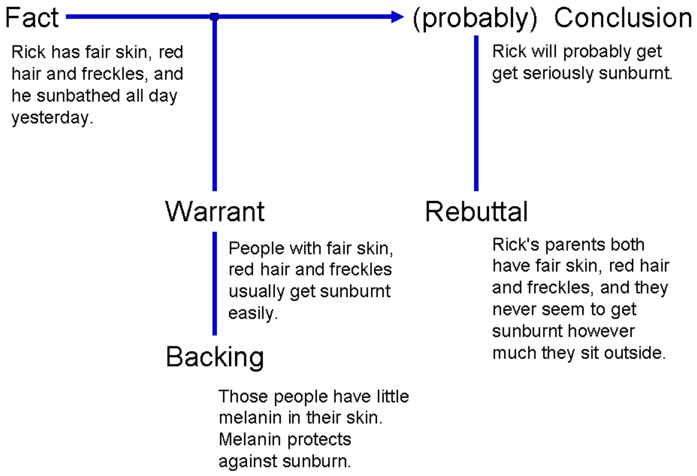
For each chain, a vector of sentence occurrence in the chain is formed. Vi= (s1i, s2i, s3i, smi). For example, LC1 appears three times in sentence one, once in sentence two, none in sentence three, and once in sentence four: V1= (3 2 0 1). Cosine similarity is calculated using an equation to measure the degree of similarity between them and put the most related in one cluster; a sentence related to the chain's highest score is extracted. Word Senses

Disambiguation (WSD) Stage is determining the sense of a polysemic word in order to ascertain the correct meaning (sense) of each word in a text based on its context.

In chapter 6 of this work, we demonstrate how to use lexical chains to capture the most important concepts of an article that contains pertinent information for warrant.

2.3.8. Toulmin Model

The Toulmin method is a technique for analysing and developing arguments that divides them into six components: claim, data, warrant, qualifier, rebuttal, and backing [155]. An illustration of this model is shown in figure 2.4.



Fact
Rick has fair skin, red hair and freckles, and he sunbathed all day yesterday.
(probably) Conclusion
Rick will probably get get seriously sunburnt.
Warrant
People with fair skin, red hair and freckles usually get sunburnt easily.
Rebuttal
Rick's parents both have fair skin, red hair and freckles, and they never seem to get sunburnt however much they sit outside.
Backing
Those people have little melanin in their skin. Melanin protects against sunburn.

Figure 2.4: The Toulmin Model [155]

Several critical components of the argumentation Toulmin model, such as warrants, may aid in improving fact checking performance when determining the veracity of a claim. We examine its efficacy and discuss it in chapter 5. As discussed in chapter 6, identifying and generating implicit warrants is critical if they are implicit.

Argument mining is "the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language" [156]. Habernal & Gurevych [157] identify argument mining as a method for analysing people's argumentation from the computational linguistics point of view and discuss the existing argumentation theories, and they develop a system based on the Toulmin model. Toulmin's arguments should be viewed as guidance to focus on the most pertinent statements and reasons for supporting or opposing the claim. It is composed of six argument components, as defined by the Toulmin Model [155]:

- **Claim:** The statement that is being argued to be true. For instance, that cat is most probably friendly.

- **Qualifiers:** implies the claim's strength from the data to the warrant and may limit the claim's universal applicability. Generally, occasionally, in most cases, frequently, few, many, it is possible, perhaps, rarely, in some cases, are all words and phrases that limit claims and are critical for determining the truthfulness of arguments. For instance, students who study more often earn more than students who study less.

- **Data:** Actual data has been gathered to substantiate the perspective (claim). It contains persuasion declarations that add clarity to the claim and demonstrate its truthfulness, such as proof, reasons, opinions, examples, and facts. For example, the following questions could be addressed: "What evidence do you have? "How did you find out? It appears to be raining, for example, the ground is wet.

- **Warrants**: a premise upon which the claim are predicated and that are often implicit[158]. The warrant will address the following: "How did you arrive at this claim based on the evidence presented, the logical connection between the data, and how did you resolve this claim."?

- **Backing**: Justification for the warrant as a more specific illustration to substantiate the warrant.

- **Rebuttals/Counterarguments**: Demonstrate an opposing viewpoint and consider other conflicting points of view. For instance, social media platforms can communicate with multiple faces using a necessary face for social needs.

For example:

- Claim: You should use social media
- Data: You have been having more trouble social lately, and over 70% of people over age 65 have social difficulty. So, social media is a good chance for elders.
- Warrant: Many social media users say it helps them to be social better (Generalization)
- Backing: 80% of social media users report a better socially and comfortable lifestyle.
- Rebuttals: 60% of old social media users suffer from a lonely feeling.
- Qualifiers: In most cases, 62% of social media users are well known in the community.

The Toulmin model is a well-known argumentation scheme that has persisted to the present day, examining the role those variant utterances may play in an argument's persuasive perspective. The following hypothesis is considered to be the connecting link between the claim and the data: if the data are correct, the claim is true [159]. Numerous works investigate argumentation-based inference mechanisms for a variety of natural language processing tasks using Toulmin's argumentation model. Gabriel et al. [160], [161] develop an argumentation-based reasoning mechanism for Belief–Desire–Intention software model (BDI) agents based on Toulmin's models. Their models are capable of generating a new belief (claim) based on available evidence and Toulmin's model's additional comments: data, warrants, and rebuttals. They also contribute by qualifying the level of confidence in a claim using the qualify function they propose in their work.

Although there has been little research on using argument analysis to detect false information, current approaches consider structured data and implicit relationships between the argument's components [162], [163]. For instance, Toulmin's model has a significant influence models used in evidence detection work [164]. Without a structured, well-defined format, data are difficult to represent using such a strict mode for false information detection that make them easily searchable, especially when considering the role of the various utterances in the argument's clear perspective.

2.4. Deep Neural Network for False Information Detection

Machine learning is a frequently used technique for automating fact-checking. The study examines a variety of machine learning algorithms for classifying false from genuine content. Utilizing various textual properties, various models train and evaluate one or more machine learning algorithms on a variety of datasets.

For Non-Neural Network Models, there are two main models used as baselines for false information detection, Support vector model (SVM) and Naive Bayes Classifier (NBC) [111], [165]. Moreover, others use different models, such as decision trees [166] and logistic regression [167].

For Neural Network Models, deep learning has demonstrated its ability to address the problem of false information, for example, using Recurrent Neural Networks (RNNs) to represent sequential posts and user engagements for twitter rumours [10], [11], [17], [102], or Convolutional Neural Networks (CNNs) to capture local features of texts and images [19], or a combination of RNN and CNN [168]. Generative Adversarial Networks (GANs) are used [20] to capture deceptive writing style features. Generative Adversarial Network with Auxiliary Classification (AC-GAN) capture more syntactic information beneficial to check the veracity of event considering Shortest Dependency Paths as the main feature [13]. [169] combines CNN and bidirectional long short-term memory (Bi-LSTM) models.

Deep learning techniques aid in the automatic discrimination of false information; deep neural networks, such as recurrent neural networks (RNNs) [56], convolutional neural networks (CNNs) [170], and, more recently, recursive neural networks [68], are extensively used for natural language processing (NLP) applications. As a starting point, we will discuss some of the most frequently used deep learning models for detecting false information.

Inspired by the human nervous system, a neural network can be thought of as a nested composite function capable of transforming data to vector representations and being trained to update the weight vector to minimise the loss function. Perceptron is a linear model of a simple neural network that can be used to solve binary classification problems [171]. Compared to a single-node Perceptron model, a neural network architecture with a collection of nodes is referred to as a Multi-Layer Perceptron (MLP). MLP comprises three distinct layers of nodes: the input layer, the hidden layer, and the output layer. It is frequently used in NLP (Natural Language Processing) applications.

Other network architectures, such as Convolutional Neural Networks (CNNs) and various Recurrent Neural Networks (RNNs), have been shown to perform well in natural language processing [172]. Neural networks convert a sequence of word embeddings into a vector for sentence representation. Numerous papers have implemented neural models for event factuality prediction [7], [13], [14]. With the rise of social media networks such as blogs, Facebook, and Twitter, neural network-based approaches, particularly deep learning models, have become the most popular for fact checking in recent years. They offer novel perspectives and methods for in-depth analysis and address the machine learning problem of relying on hand-crafted features by extracting hidden features and representations in the text to determine the credibility of information. Chapter 4 and the subsequent chapters make extensive use of the deep learning models discussed below.

2.4.1. Convolutional Neural Networks (CNN)

CNN's use convolutional layers to extract the most advantageous local patterns from the input and pooling (sample reduction) layers to mitigate the overfitting problem by reducing the number of parameters connected to the final fully connected layer via activation functions. In addition, CNN is used to classify sentences, where several filters in pooling layers are used to address sentence length variation [173].

When using pre-trained static word vectors for classification tasks with CNN, Niven and Kao [174] demonstrate that pre-trained word vectors perform extremely well. They provide an overview of the general approach to applying deep learning techniques to natural language processing. First, sentences are transformed into embedding vectors and fed into the model as a matrix. Convolutions are performed word byword across the input using various kernel sizes, such as two or three words at a time. Following that, the resulting feature maps are condensed or summarised using a max-pooling layer. Kim includes a diagram in figure 2.5 that illustrates the sampling of the filters by using different-sized kernels as distinct colours (red and yellow).

CNN is capable of capturing critical local information but discards useful long-distance relations between words. By considering windows during training, previous methods demonstrated that CNN is robust at detecting unigram features. A CNN enhancement that uses semantic filters to weight initialises convolutional filters rather than randomly initialise filters with significant n-grams. For instance, in the case of uncertainty: "it may rain, so take care," an "may rain" n-gram is more critical than the rest of the sentence. Naive Bayes has been used to identify the most critical words and apply various equations [175]. The authors found CNN useful in [168] for determining the credibility of propagated information. Recently, a model based on convolutional neural networks (CNNs) and probabilistic weighted average pooling were proposed for automatically handling negation and speculation texts [176].

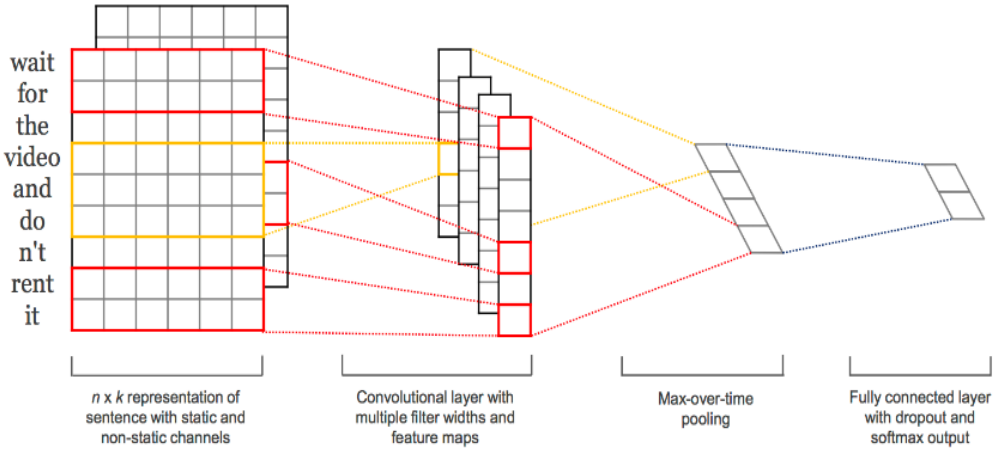

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 2.5: An example of a CNN filter and polling architecture for Natural Language Processing [173]

Graph Convolutional Networks (GCN) is a very powerful CNN architecture for machine learning on graphs to produce useful feature representations of nodes in networks. GCN is a neural network capable of handling arbitrarily structured graphs, convolutional layers, and fully connected layers representing data in the form of graphs [177]–[179]. Each state updates its state using information from its neighbours, and each convolutional layer manages first-order neighbourhood information [179], [180]. Hu et al. [181] propose the

Multi-depth Graph Convolutional Networks model, which explicitly preserves the multi-granularity of information. This approach can increase classification accuracy and provide a more precise understanding of the nature of news than currently available techniques. In our work, we use CNN for Discriminator to discriminate between fake and real arguments in chapter 7 for capturing local features of texts and enables the generation of rich feature representations for individual sentences based on contextual and salient information.

2.4.2. Recurrent Neural Networks (RNN)

A Recurrent Neural Network (RNN) represents an entire sequence by calculating the previous time step with input to represent the current time step hidden state and feeding it to activation functions; this concept is derived from the feed-forward neural network [101]. Although RNNs deal with sentences of varying lengths, they maintain weights throughout the training process for long texts. These issues are gradient vanishing and exploding. Long-Short Term Memory addresses this issue. Three gates control the flow of information in an LSTM cell; the input gate determines how much data is provided to the cell memory state, the forget gate determines how much data is ignored, and the output gate determines how much data is output. The architecture of LSTM and its implementation facilitate the handling of lengthy steps [182]. Bi-directional RNNs can process sequences in both directions (forward and backwards) and concatenate them to obtain additional contextual information. The works in [11], [141] used RNN deep learning to detect fabricated data. The LSTM layer captures contextual information and merges forward and backward features using element-wise summation, resulting in richer semantic information, as in equations 2.5-2.12, where $\oplus$ denotes concatenation and $\otimes$ denotes element-wise multiplication.

Equation 2.5 $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$

Equation 2.6 $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$

Equation 2.7 $g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

Equation 2.8 $c_t = i_t \otimes g_t + f_t \otimes c_{t-1}$

Equation 2.9 $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

Equation 2.10 $h_t = o_t \otimes \tanh\,(c_t)$

Equation 2.11 $h_i = \overrightarrow{h_i} \oplus \overleftarrow{h_i}$

Equation 2.12 $H = [h_1, h_2 \ldots\ h_t]$

The hidden layer state $\overrightarrow{\mathrm{h}_{\mathrm{i}}}$encodes the information features in the forward direction, while the hidden layer state $\overleftarrow{\mathrm{h}_{\mathrm{i}}}$ encodes the information features in the backward direction. W, b are model parameters to be learned and $x_t$is a word of sequence. Hidden state ht−1 and the input at the current time step xt: the forget gate ft determines the degree to which an existing memory in the old cell state is kept in the cell new state, the input gate signifies the degree to which that information is added which influences the cell state, and the output gate it controls the amount of information stored. In the new cell, the state is used to compute the hidden state of the LSTM unit. These gates collectively decide how to update the current memory cell ct, updated with new information computed according to part of the existing memory and new values and the currently hidden state ht.

For BiLSTM, obtain the summarised representation in both directions to represent the word sequence while obtaining contextual information about the current word and learning long-term dependencies. Additionally, we use them to encrypt the sentences. BiGRU is a variant of BiLSTM that simplifies the gating mechanism and performs well during training, obtaining text features quickly. We use it to gain knowledge about CNN by analysing the most salient data and identifying significant local features. While CNN lacks global and long-distance features, it is faster to train.

Bi-GRU illustrated its efficacy in encoding document representations, and Yang et al. [91] demonstrated the hierarchical attention mechanism's ability to optimise document representations. Additionally, as Zhang et al. [183] demonstrate, it is robust for cross-domain sentiment classification. Finally, Gao et al. [184] demonstrate that employing the attention mechanism improves the accuracy and speed with which CNN-based models capture the internal structure of sentences and learn linguistic patterns. Recurrent Neural Networks is shown in figure 2.6.

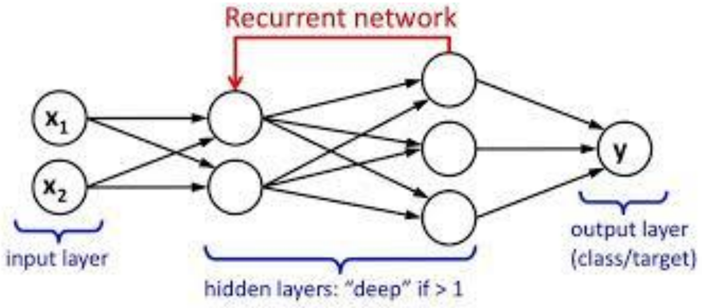


Figure 2.6: Recurrent Neural Networks (RNN) [101]

In our work, we use a Bi-LSTM in different places, e.g., to generate warrants as in chapter 6 to obtain contextual information about the current word by reading each sentence in two directions: forward (forward) and reverse (backwards). The final encoded representation is a composite of the Bidirectional hidden state and the Bidirectional hidden state representations.

2.4.3. Variational Auto-Encoder (VAE)

In VAE, the encoder converts input data to latent vectors via multiple hidden layers, and the decoder then reconstructs the latent vectors back to the original data [185]. The variational auto-encoder performs the function of a generative model [186]. It aids in the generation of compressed feature vector representations and is robust due to the distilled information it contains.

VAE is a hybrid autoencoder/variational inference technique in which the latent code learns the probability distribution for the new data and then uses it to generate new data. The model employs a variational autoencoder with latent space for each generator [187], as illustrated in figure 2.7, where the style code is c, and the left latent space is z. (remaining information). Meanwhile, a discriminator evaluates the likelihood that the text transferred was written by the target semantic domain. CycleGAN is a neural network that employs two conditional generators and two discriminators. Each GRU encoder was conditioned on the output of two second-level encoders: one for the style of other domains and another for the content of other domains. X is the original text, and x-hat is the generated text.

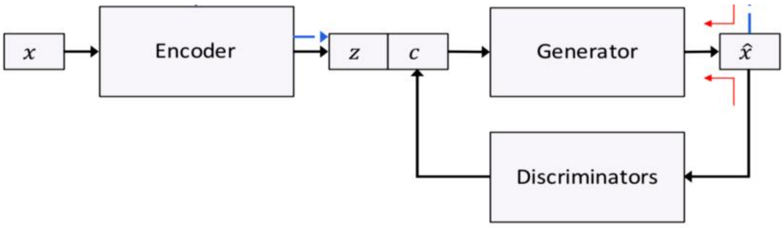


Figure 2.7: Variational autoencoder with latent space [187]

VAE is an unsupervised generative text model that extracts a continuous semantic latent variable for each piece of information. VAE models have been proposed [188], consisting of an encoder that converts each sample to a latent representation and a decoder that generates samples from the latent space. In addition, conditional VAE [189] was proposed to generate more diverse and relevant text that outperforms seq2seq models. Seq2Seq is a training model for converting sequences of items e.g., words, letters, and so on and producing another sequence of items. Other models incorporate additional variables representing the generated text as an additional input to the decoder using Conditional VAE [190]–[194].

Padnekar et al. [195] propose a new model for predicting fake news stances that combine Bi-Directional Long Short Term Memory (Bi-LSTM) and Autoencoder. Context is learned within this model via the two Bi-LSTM blocks for the headline and body word embeddings. After concatenating the encoded feature with other features, it is fed into an autoencoder. The autoencoder compresses the higher-dimensional feature vector, reducing the data's complexity and reducing the number of dimensions. The dense layer that follows provides critical context for the relationship between the headline and the body of the article. In our work, we use Variational Autoencoder Model for claim generation because it works better at forcing the decoder to use latent vectors and can perform significantly better than other language models such as GRU as in chapter 5.

2.4.4. Generative Adversarial Network (GAN)

Generative adversarial networks (GAN) is a novel method for developing generative models based on deep learning, it has the potential to overcome some of the shortcomings of some traditional generation models in practise and to subtly optimise some loss functions that are difficult to deal with via adversarial learning [196], [197]. GAN is critical in determining the overlap distribution to use in estimating distances between clusters/networks of data. Two populations (or samples) are said to be similar if their distribution functions overlap.

Recent works try to solve GAN's main problems like the models never converge, limited varieties of samples, the generator learns nothing due to the discriminator's high accuracy. Vani used reinforcement learning (policy gradients) to train a generator based on the discriminator and actor-critic framework. The generator receives a signal from the discriminator tries to produce trajectories samples invariant from real from the expert policy. Compared to SeqGAN, in which signal comes after a whole sequence based on Monte Carlo policy, the action-value function is used at each timestep (local level) for estimation. They inferred the WGAN objective to minimize Wasserstein distance between the expert policy πε (discriminator), which has real trajectories and that of the learned policy π (generator models), which has sample trajectories [198]. A categorical distribution, according to previously generated tokens, is produced by the generator. The actor updates its parameters to get better results of the critic's policy [198]. State-action value in conditional sequence GAN is implemented where input x, generated prefix yˆ1:t and z are generated sequence conditioned to x and y [199].

The Generative Adversarial Net (GAN) employs two models: a discriminative model that directs training by distinguishing between real and unreal data and a generator model that captures data distributions that match either marginal or conditional distributions [196],

[197], [200], [201]. GANs are critical for determining the distribution of overlap. GAN may be able to identify critical rather than specific characteristics [20].

In continuous (rather than discrete) output, generative models are trained by back-propagating gradients from a discriminator to the generator. Different training methods such as reinforcement learning are used to guide generative models in the case of discrete output. GAN is extended to take into account the discrete nature of texts [100], [202]–[205] for a variety of natural language processing tasks, including commitment detection in email [206] and protein-protein and drug-drug relationships [207].

Due to the discrete nature of the textual data, no gradient can be obtained, and no backpropagation from the discriminator to the generator used to train it will occur. Different approaches rely heavily on deep networks that use a variety of technologies to guide the generator and learn how to minimise the loss cost associated with a function. Vani used reinforcement learning (policy gradients) to train a discriminator-based generator while incorporating an actor-critic framework. The generator receives a signal from the discriminator and attempts to generate trajectories samples that are invariant from the expert policy's real trajectories.

2.4.5. Attention-Based DNN Models

Attention-based DNN Models concentrate on the most salient parts of the source sentence and provide a complete representation. We believe it is beneficial to practise it and observe its leverage in this research. We will demonstrate some recent attention based DNN models in this section.

There are variant architectures to apply attention mechanism [142], [143], [184], [208]. Hu surveyed different attention mechanisms: basic attention to retrace relevant elements from a sequence, multi-dimensional attention that are used to extract interactions of terms, hierarchical attention to detect global and local features, self-attention focus on contextual information, memory-based attention to capturing hidden dependencies and task-specific attention which pay attention to relevant information related to a precise task used in sequence-to-sequence models [208]. The authors in [184] propose Attention-Based Bidirectional Long Short-Term Memory Networks (Att-BLSTM) for detecting relation classification and the necessary information for this detection without relying on extracted features from lexical resources.. The hidden vectors from LSTM are used in the attention mechanism to represent a sentence [184]. Gao et al. developed the hierarchical structure of CNN with an attention mechanism called Hierarchical Convolutional Attention Networks.

RNN extracts features from an entire sequence, whereas CNN deals with sentences part by part, as is well known (windows). The goal of combining attention with CNN is to get the benefits of CNN's fast training and RNN's advantage of learning linguistic relationships over long sequences [143]. Yi et al. proposed a bidirectional recurrent neural network with a multiple attention layer model, word-level attention: extracting the most critical features to represent the sentence, then generating an output vector at the sentence level. Attention is used to pick up important details from other relevant sentences [142]. Yu et al. [209] investigate the effect of CNN's attention mechanism on the learning of latent textual representation by extracting temporal and semantic representation of events. For the prediction of a target, it provides important weights of elements in an input sequence. GRU, LSTM, or other networks are used to first encode an input sequence.

Equations 2.13-2.18 used for computing an attention-weighted vector for the i^{th} element in an input sequence is given by Yang et al. [91]. The hidden state hj represents an encoder state at time step j for (j = 1, 2, …, T).

Equation 2.13 $\vec{h}_{ij} = \overrightarrow{LSTM}_{(\widehat{w}_{ij})}; \quad i \in [1, C], j \in [1, N_i]$

Equation 2.14 $\overleftarrow{h}_{ij} = \overleftarrow{LSTM}_{(\widehat{w}_{ij})}; \quad i \in [1, C], j \in [N_i, 1]$

Equation 2.15 $h_{ij} = \vec{h}_{ij} \oplus \overleftarrow{h}_{ij}$

Equation 2.16 $u_{ij} = \tanh\left(W_w \cdot [h_{ij}] + b_w\right)$

Equation 2.17 $a_{ij} = softmax(u_{ij}) = \frac{exp(u_{ij})}{\sum_{t=1}^{N} exp(u_{it})}$

Equation 2.18 $c_i = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij}$

The context vector ci is defined as a weighted sum of hidden states h of the input sequence, weighted by attention scores a. where $u_{ij}$is attention weights, a_{ij} denote a score function and ci, are attention-weighted sequence vector (Yang et al., 2016a), respectively. Wh and bh are randomly initialised weights and biases.

Ren & Zhang [210] employed a novel hierarchical attention mechanism to detect fake news by classifying news article nodes in the heterogeneous information network, considering both node-level and schema-level attention to learn the comprehensive representations of news article nodes. Attention-based DNN Models focus on relevant parts of the source sentence and provide richer representation. In this research, we believe it is valuable to practise it and notice its leverage. In this section, we will show some recent attention based DNN models.

Attention-Based Bidirectional Long Short-Term Memory Networks (Att-BLSTM) is implemented to detect the necessary and relevant information by applying equations 2.19-2.21 [211]:

Equation 2.19 $M = tanh\ (H)$

Equation 2.20 $a = softmax\ (w^T M)$

Equation 2.21 $r = H * a^T$

Gao et al. [184] developed a hierarchical structure of CNN with a Hierarchical Convolutional Attention Networks attention mechanism. It is known that RNN extracts the features from an entire sequence while CNN deals with sentence by part (windows). The mixture of attention with CNN is to get the benefit of fast training from CNN and the advantage of learning linguistic relationships over long sequences as in RNN. Scaled Dot Product Attention: Comparing each input embedding word with the word embeddings sequence to obtain a relationship. Convolutional Multiheaded Self-Attention: implementing multiheaded (parallel) attention for portions of embedding instead of single attention for all dimensions of the embeddings and finally concatenating produced heads from the individual scaled dot product attention. Convolutional Multihead Target-Attention: the comparison will be made on a learnable target vector instead of each entry and Positional Embeddings to find the order of words in a sequence [184].

Yi et al. [143] proposed a bidirectional recurrent neural network with multiple attention layer model to extract Drug-drug interaction (DDI), Embedding Layer: a combination of features that represents dimensional space of word embedding and position embedding, then sentence, then feed the output to RNN layer. Bidirectional RNN Encoding Layer: to read words of a sequence by RNN's gated recurrent unit (GRU). Word Level Attention: to extract the essential features to represent the sentence, then get output vector such Sentence Level Attention is implemented to capture essential features from other relevant sentences [143].

Hu [208] surveyed different attention mechanisms: primary attention to retrace relevant elements from a sequence, multi-dimensional attention that ale to extract interactions of terms, hierarchical attention to detect global and local features, self-attention focuses on contextual information, memory-based attention to capturing hidden dependencies and task-specific attention which pay attention to relevant information related to a precise task used in sequence to sequence models [208]. In our project, In Chapter 7, we employ an attention mechanism to generate the Toulmin argument by utilising the most pertinent segments of the input sequence.

2.4.6. Multi-Head Attention

According to Vaswani et al. [212], "multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.". Multi-head is applied to compare each hidden state with other hidden states in the same layer as equations 2.22 and 2.23:

Equation 2.22 $h_i^1 = f_{self-attn}^1 \left(e_i, [e_1, e_2, \ldots, e_N]\right)$

Equation 2.23 $h_i^{l+1} = f_{self-attn}^{l+1} \left(h_i^l, \left[h_1^l, h_2^l, \ldots, h_N^l\right]\right)$

Multi-Head Attention-based CNN or Bi-LSTM (Self-multi head attention: take advantage of self and multi-head attention and Self-attention) benefit from obtaining the internal spatial relationship in words represent long text. The attention module of the encoder that we use is mainly based on multi-head attention [212]. We use self-attention to compute a representation of the sequence, which is an attention mechanism relating to a single sequence's different positions. Output H is Query vectors, keys vectors and values vectors, the attention distribution a^t is calculated as equations 2.24-2.26:

Equation 2.24 $e^t = \frac{Q^e K^{eT}}{\sqrt{D}}$

Equation 2.25 $a^t = softmax(e^t)$

Equation 2.26 $Attention(Q, K, V) = a^t V^e$

The multi-head attention adjusts the Q, K and V matrix dimensions by different linear layers to the dimension's queries, keys, and values. Thus, the linear transformation parameters (W) of Q, K and V, are different each time based on the learnable parameter's matrix for the heads. Then h parallel heads are employed to focus on different semantic spaces.

2.4.7. Capsule Network

Instead of using a single neural node as in CNN models, Hinton et al. and Sabour et al. [213], [214] proposed a capsule model that uses a neuron vector for the input and output layer with a dynamic routing algorithm instead of pooling operations. It's used in tasks like question answering [215], word segmentation [216], and extracting global semantic features of different categories [217], sentiment analysis [218]–[220], cross-domain [220], [221], sarcasm detection [222], and propaganda detection [223] to understand spatial information and learn contextual information of text. According to Gao et al. [184], the capsule network can extract a more detailed representation of a text and other important features such as word position, semantic structure, and syntactic structure.

In our model, we apply Gao et al.'s equations [184] to obtain the capsule network's output, and we consider other output of various types of encoders such as CNN and BiLSTM. The outputs of capsules networks are achieved in equations 2.27- 2.29.

Equation 2.27 $\hat{\mathrm{u}}_{\mathrm{o|i}} = \mathrm{w}_{\mathrm{io}}\mathrm{h}_{\mathrm{in}}$

Equation 2.28 $\mathrm{S}_{\mathrm{out}} = \sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{c}_{\mathrm{io}}\, \hat{\mathrm{u}}_{\mathrm{o|i}}$

Equation 2.29 $\mathrm{c}_{\mathrm{io}} = \frac{\exp(\mathrm{b}_{\mathrm{io}})}{\sum_{\mathrm{k}} \exp(\mathrm{b}_{\mathrm{ik}})}$

Where $\boldsymbol{c_{io}}$ is a coupling coefficient, is determined by the dynamic routing method. And $\boldsymbol{S_{out}}$ The activation of capsule network output is calculated by the nonlinear function (Squash) for normalization purposes, as shown in equation 2.30.

Equation 2.30 $v_{out} = \frac{\|S_{out}\|^2}{1+\|S_{out}\|^2} \frac{S_{out}}{\|S_{out}\|}$

Where $\boldsymbol{v_{out}}$ is the output vector of the capsule network. The dynamic routing method [220] is shown below in equations 2.31-2.33:
for all capsule i in layer l and j in l+1:
initial: $\boldsymbol{b_{ij} \leftarrow 0}$
for iterations do

Equation 2.31 $c_{ij} \leftarrow soft\,max(b_y)$

Equation 2.32 $s_j \leftarrow \sum_i c_{ij} u_{j|i}$

Equation 2.33 $v_j \leftarrow squash(s_j)$

for all capsule i in layer l and j in l+1: $\boldsymbol{b_{ij} \leftarrow b_{ij} + u_{j|i} \cdot v_j}$, return $\boldsymbol{v_j}$

The architecture of the proposed RNN-based capsule model is shown in figure 2.8, where Ns is the number of words H = [h1, h2, . . ., hNs] is the hidden vectors of an input instance encoded by RNN. The instance representation vs is the average of the hidden vectors. All capsules take the hidden vectors as input, and each capsule outputs a state probability pi and a reconstruction representation rs, i.

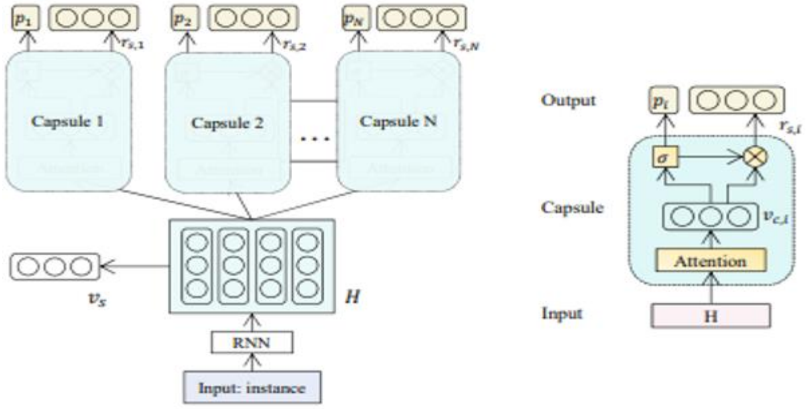


Figure 2.8: The architecture of RNN-Capsule [224], the architecture of a single capsule. The input to a capsule is the hidden vectors H=[h1, h2, ..., hNs] from RNN.

2.4.8. Deep Reinforcement Learning

The reinforcement learning model uses reward signals to solve complex machine learning problems [225]–[227]; In general, when dealing with discrete data, the agent is rewarded for taking discrete actions by the policy that directs the actions. After reaching the end of the sequence, the accumulated reward is calculated. The internal state of the policy model is updated, and the agent chooses a specific action and observes a reward for that action, which is optimised by policy to maximise the expected discounted reward. The following are the Reinforcement Learning Components, as in the following figure 2.9.

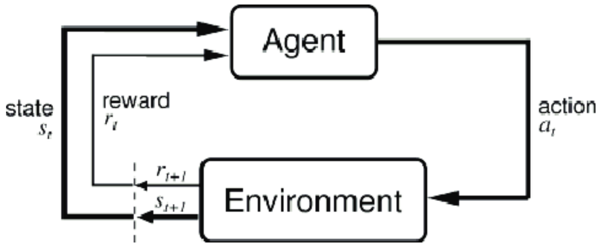


Figure 2.9: The agent-environment interaction in reinforcement learning

- Agent: It is an entity that determines which action to take to maximise a reward in a given environment.
- Environment (e): A place with which an agent can act upon.
- Reward (R): An evaluation of an agent's behaviour (actions).
- State (s): It is the state of the environment at the moment.
- Policy (π): The procedures followed by an agent in determining the most rewarded action.
- Value Function: represents the long-term reward achieved by beginning with that state and implementing a specific policy. A future (discounted) reward is expected to occur due to the policy from the initial state.

We employ reinforcement learning agents, such as the Deep-Q Network (DQN), in our model to help generate more informative (attention) features and to correct the decoded generated output for the generative model by enriching the GAN model with additional contextual information. Numerous reinforcements learning agents were used in our experiments to train the generator by feeding it with correct representations.

Hierarchical Reinforcement Learning HRL is based on policy hierarchies, similar to how deep learning is based on feature hierarchies. In early work in this area, strategies were introduced to allow policies to execute additional policies and primitive actions. This strategy allows top-level policies to concentrate on higher-level objectives, while lower-level policies charge sub-objectives [228]. Chapter 7 employs HRL to generate factual assertions. A hierarchy of policies is used to share knowledge across generation subtasks in order to optimise performance.

2.4.9. Domain Adaptation

Domain adaptation is the process of identifying common shared characteristics and domain-specific characteristics between the source and target domains and excluding them from the train and test samples [206]. For the current research, unsupervised domain adaptation using adversarial networks resulted in a representation indistinguishable from the source or target domains by the discriminator, thereby generating domain invariant features using adversarial learning methods. Previous works proposed both generative and discriminative approaches to accomplish this [128], [229]–[234]. The most frequently used approach for the NLP task is the model proposed in [233], [235]–[240]. However, because the representation strategy used to transfer source domain knowledge is critical in the affective domain adaptation model, the task becomes more difficult as the domain discrepancy grows [241]. To address this issue, some researchers employ distribution strategies to capture more invariant features [242], reduce the distance between source and target representations [243], or maximise domain discriminator loss to learn shared features [233], [244]–[247].

Azarbonyad et al. [206] examined Commitment Detection in Email at the feature and sample level. They determined which domain-specific features are the most discriminating and should be eliminated in favour of equivalent features. Importance sampling is used to estimate parameters for the target distribution using the data from the source distribution. We introduce a novel domain adaptation technique for detecting false information in chapter 4.

2.5. Summary

This chapter discussed the history of false information detection. In section 2.1, we introduced various definitions of false information and research efforts to characterise it on social media. The task of detecting false information is frequently framed as a classification problem. In section 2.2, the methods for detecting false information were detailed. In section 2.3, several critical concepts such as Toulmin's model, RTS, language model, and other related concepts, as well as Deep Neural Networks for false information detection, were introduced in section 2.4. The subsequent chapters will detail the methods and experiments used to accomplish the objective.

Chapter Three: **Related Works**

3.1 Introduction

In general, the main pipelines for false information detection are rumour detection (to determine whether or not the claim is based on rumour, which may originate in social media, a political discussion, or other sources), evidence retrieval (to retrieve evidence from the Web, social media, Wikipedia, or a knowledge base), stance detection (determining whether the author of a piece of text is for or against a given target) and veracity prediction (fact checking or truth discovery) to determine the veracity of the claim. False Information Detection Pipelines is illustrated in figure 3.1. This chapter discusses related works on the detection of false information. We will typically discuss recent advances in the fundamental tasks of false information detection: stance detection and veracity prediction since we suppose that people's initial reactions to an emerging claim can be indicative of its veracity.

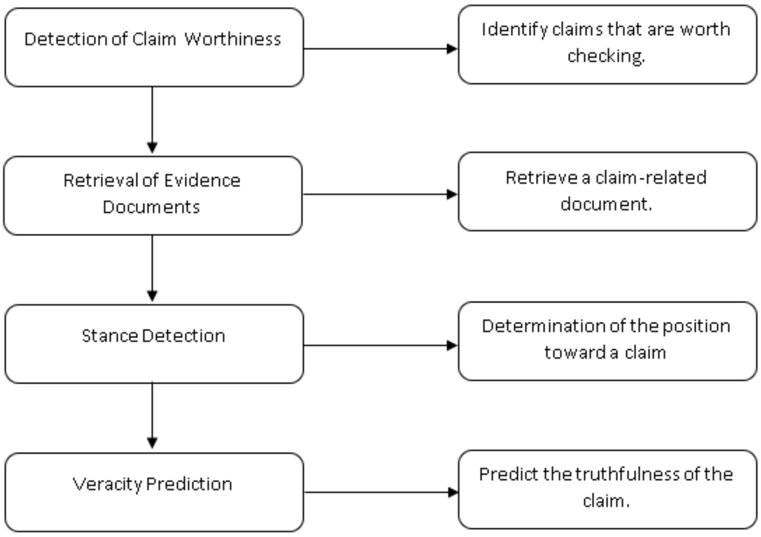

Detection of Claim Worthiness
Identify claims that are worth checking.
Retrieval of Evidence Documents
Retrieve a claim-related document.
Stance Detection
Determination of the position toward a claim
Veracity Prediction
Predict the truthfulness of the claim.

Figure 3.1: False information detection pipelines.

Unlike fact checking, which is concerned with determining the truthfulness of a claim given a specific piece of evidence, and unlike truth discovery, which is concerned with determining the actual true fact of a claim based on multiple conflicting stances from variant documents [248], the primary task of rumour detection is to classify post statements as rumour or non-rumour, where rumour is unverified information at the time of posting and it is a deliberate fabrication [107], [249]. The current information is unconfirmed by known specialists and is primarily spread via social media platforms such as Twitter and Facebook, which negatively impact individuals and society. Rumour detection is a critical pre-stage in determining the veracity of rumours (fact checking); if a rumour is detected, its veracity is determined. The primary objective of rumour detection is to prevent users from spreading this type of false information [250]. Rumour detection is also an essential task in fact checking, as it identifies statements that should be verified, and if they are rumours, the verification process should be followed.

Numerous comprehensive surveys [23], [249], [251], [252] have been conducted, with the majority of them relying on supervised models and requiring significant engineering effort [106], [253], [254]. Recent work has used deep learning to address the limitations of feature engineering, such as using recurrent neural networks (RNN) to automatically learn the representations of posts [11] and a hierarchical attention model [81] to detect false articles [17].

Zubiaga et al. [249] defined a rumour classification system as having four components: rumour detection, rumour tracking, rumour stance classification, and rumour veracity classification. Zhao et al. [255] used A cluster ranking algorithm that ranks tweet clusters according to their likelihood of being rumoured. Zubiaga, Wong Sak Hoi, et al. [256] implemented the unsupervised method using a sequential classifier that learns lexical and temporal features of rumours.

This work discusses briefly several models that are relevant to the primary objective of this work. It begins with stance classification and veracity checking, progresses to joint stance and veracity prediction, and concludes with the discovery of related datasets.

3.2 Stance Detection

Stance detection is the classification of an individual's attitude toward a target expressed in a text, e.g., agree, disagree, discuss, or unrelated, based on evidence from this article [257]. It emphasises opinions and diverse perspectives to promote a better understanding of contentious issues [3], as demonstrated in debates [158], Twitter posts [258] or online discussions [259]. Stance detection can be advantageous as a pre-processing stage's believability feature and can be applied to fact checking[114], [260]–[262], rumour classification [263] and veracity checking [249], [264]**.**

Other methods for stance detection have been proposed, including those based on linguistic features [265], [266], and hand-crafted lexicons [16]. Recent research demonstrates that competitive performance in detecting a document's stances toward a specific claim can be obtained using a neural network, such as memory networks [6] with conditional LSTM encoding [267], attention mechanisms [268], Bidirectional Encoder Representations from Transformers [269], gradient-boosted decision trees (GBDT) model, and convolutional neural networks (CNNs) [5]. Additionally, the adversarial domain adaptation technique is used to compensate for the limited size of labelled data, particularly across domains [261], [270]. Mohtarami et al. [6] used a memory network and achieved 61.67 macro F1 scores on the Twitter dataset where Macro-F is calculated by arithmetic mean of all per-class F1 scores. Their models encode the claim and its source using distinct DNN models and then assign the correct prediction using a combination of similarity matrix and memory network. These systems made extensive use of hand-engineered features such as TF-IDF, Singular Value Decomposition (SVD), Word2Vec, and sentiment, while the second extracts feature such as unigrams, latent Dirichlet allocation, latent semantic indexing, and topic models.

Concerning the FNC dataset training models, stance detection measures the degree of similarity between a claim and evidence. Here we will examine the top three ranks of models trained on the FNC dataset. Baird et al. [5] are ranked first, with an F-score of 82.02; their system combines a decision tree model with a deep CNN model. The decision tree model extracts feature such as count, sentiment, and others. Contextual information is gathered by embedding words into the CNN, with the final SoftMax layer supplying the detected classification results. The final prediction is derived from the combination of the two models. With an F-score of 81.97, Hanselowski's UCL Machine Reading [271] comes in second place, where latent Dirichlet allocation and latent semantic indexing are the primary features used in training. Riedel et al. [4] are ranked third with an F-score of 81.72; TF and TF-IDF. The hidden layer receives the features, and the SoftMax layer outputs the result; they trained their system solely on the most frequently occurring terms using cosine similarity. The relative score evaluates a model by dividing the stance detection task into two sub-tasks: related/unrelated classification and agree/disagree/discuss classification [272], as illustrated in table 3.1.

On Emergent data**,** Ferreira and Vlachos [257] proposed a logistic regression model that uses news headlines' lexical and semantic characteristics to predict whether a claim is for, against, or discuss. Table 3.1 summarises some state-of-the-art models that compare their model to various baselines for the stance detection task. Finally, in chapter 8, we will compare our findings to those of Zhang et al. [272] on Emergent data, The relative score evaluates a model by dividing it into two subtasks: related/unrelated classification and agree/disagree/discuss classification.

Table 3.1: A summary of stance detection related work on Emergent

The model	The implementation details	Emergent Relative Score
LSTM (BiLSTM)	Stance Detection with Bidirectional Conditional Encoding. The encoded claim is used as initial states to encode the evidence [267] after the 100-d GloVe word embedding is applied [146]	87.69
Attentive CNN (AtCNN)	For both claim and evidence feature representations, the convolutional neural network is used and attention mechanism to extract the most relevant features [273]	83.56
Memory Network (MN)	A combination of convolutional and recurrent neural networks by an end-to-end memory network is implemented [6]	85.92
Ranking Model (RM)	Ranking model to maximise the difference between the four stances representation agree, disagree, discuss, unrelated [274]	87.69
Official Baseline	gradient boosting decision trees model for stances [275]	74.86
Logistic Regression (LR)	After checking whether the source is related or not by n-gram matching and rule-based methods. The stances: agree, disagree and discussed are decided by Logistic Regression [276]	83.45
Gradient Boosted Decision Trees (GBDT)	Apply Gradient Boosted Decision Trees to detect related stance and apply another Gradient Boosted Decision Trees to detect the remaining three stances [4]	87.53
Multi-Layer Perception (MLP)	Cosine similarity between claims and evidence, and Multi-Layer Perception for the four stances [277]	85.43
Hierarchical representation of a neural network	Hierarchical representation of these classes combines agree, disagree, and discuss classes under a new related class where the hierarchical architecture alleviates the class imbalance problem. One neural network layer for related stance detection and the second layer is for the three stances detection [272]	89.30

On Rumour Eval 2019, stance detection has been demonstrated to be a critical task for rumour verification in a variety of studies [107], [255], [278]–[284], and some studies, for example [285], proposed a multi-task learning framework for jointly predicting rumour stance and veracity. Numerous models were entered in the 2019 Rumour Eval competition [286]. In this competition, Yang et al. [287] report the best performing system for the stance detection task using the inference chain of conversation from source post to replies. The system is dependent on features such as the number of question words, the presence of BiLSTM and Transformer rumour words, incorrect synonym and false antonym. The second-best system is by Fajcik et al. [288], which employs an ensemble of BERT, and the third-best system is by Baris et al. [289], which employs pre-trained representations with OpenAI GPT. Pre-training representation models [290] and ELMO [147] have demonstrated promising results in which each word's representation is based on its use context.

Due to a scarcity of labelled training data for the target, the majority of current research predicts the stance without encoding the target, allowing the stance to be misinterpreted as belonging to another target. Additionally, they separate stance detection and fact checking, despite the fact that stances can aid in determining veracity. Meanwhile, the purpose of this work is to develop stance classification models that consider the intended audience for the claim. Consequently, we introduce a novel model for aggregating stances to validate the claim, as well as a novel multi-task learning model in chapter 9 to address the aforementioned challenges.

3.3 Fact Checking

The term "fact checking" refers to the process of determining the veracity of claims through the use of evidence. There is a rich literature on fact checking that aims to measure the truthfulness of a claim for the given evidence [9], [95], [117], [291]–[296]. Vlachos & Riedel [9] released 221 labelled claims in the political domain; they consider intermediate classes as "mostly true" or "half-truth" when the sentences are not entirely fake or real. Mitra & Gilbert [291] labelled a dataset of approximately 60 million tweets about more than 1,000 news events according to their credibility. Samadi et al.'s approach [292] jointly estimates the credibility of sources and correctness of the claims using the Probabilistic Soft Logic framework. Wang [117] released a dataset collected from fact checking website PolitiFact; labelled by multiple classes: pants-fire, false, barely true, half-true, mostly true, and true. Nakov et al. [293] fact check shared task on automatic identification and verification of claims in political debates, automatically estimating the check-worthy claims' level of fact checking. Hanselowski et al. [95] proposed a method to generalise unseen data to deal with Fake News Detection problems. Karimi et al. [294] applied a multi-class fake news detection framework where a combination of LSTM, CNN, and a fully connected network to determine the veracity of fake and real where they integrated multiple pieces of information about a claim. Alhindi et al. [295] extended the LIAR dataset and labelled a claim, and they use meta-information and "justification," human-written reasoning for factuality checking. Yin & Roth [296] address textual entailment of detecting false claims and prove joint learning could enhance both tasks: claim verification and evidence selection.

Most veracity checking systems have been developed over FEVER [8]. FEVER is a large-scale dataset for fact extraction and verification that consists of 185,445 claims and their related evidence. The best performance on the first FEVER shared task recently is the Bi-Directional Attention Flow (BiDAF) network [297], Neural Semantic Matching Networks (NSMNs) [298] and the contextualised representations of a pre-trained BERT [269] as in Soleimani et al.'s [299]. Seo et al. in BiDAF [297], two vector sequences are produced from the embedding layer for both claim and evidence, and the attention scores are computed by the attention layer, which sends them to the output layer where the semantic similarity between the original sequences and the new vectors is computed. Finally, the label is yielded by the output layer. Nie et al. in NSMNs [298], the alignment layer is applied for the encoded claim and evidence sequences then semantic matching is performed by an LSTM matching layer, where the output is sent as input to the output layer to produce a label. In BERT, 12 encoder layers with self-attention with a classification layer are applied to get a highly embedded representation of the claim and the evidence; the classification layer receives this representation to output labels.

Recurrent/Recursive Neural Networks RNNs are used to represent posts and user engagements [10], [11], [17], [18] in this data. Convolutional Neural Networks (CNNs) to capture local features of texts and images [19] are applied in medical texts [168]. This study uses accurate pictures to assess reliability. To combine image processing with text, separate tasks and combined datasets are required. If the image is real and does not violate the rules of the paper. Generative Adversarial Networks (GANs) have been applied to produce fake news, and it has been used to gain a "general feature set" for fakes across events [20].

Ma et al. [10] investigated fake news propagation by a recurrent neural network based on user properties and fake news propagation profiles based on users' information. The risky in this system is to which extent the user profile is real and not fake, or there are different reasons behind creating this profile and opposing opinions or comment as "fake deny" or "fake support". Ruchansky et al. [17] developed CSI model specification with three components: Capture module based on LSTM to get textual information of the pattern of temporal engagement to an article, while the Score module extracts source characteristics for all users, the combination is done between article representation which comes from the first module and user information representation that comes from the second module, they are combined in integrating module to classify the fakes news.

A recurrent neural network (RNN) is used to extract the links between news creators and their subjects [18]. LSTM is used to reveal the representation of temporal textual characteristics of rumours on Twitter in real-time. Even this system can learn without training, access hand-crafted features, or easily discover hidden representations [11].

Multiple convolutional layers are used to merge the inputs from the text and the image into a common representation. This model performs well but requires substantial data to train. Liu and Wu [168] proposed an early detection system by training a combined CNN and RNN to categorise false claims post instantaneously.

It is noticed that DNN models are mainly used, but some of these systems have a computational limitation, e.g. in Karadzhov et al. [85], where retrieving evidence to compare sometimes taking a long time, especially the filtering process. Some of these systems require continuous observation of post changes, but other systems can be trained only through supervised data [10], [11], [17], [18]. Wang et al. [20] used text and visual features to train their models. There is a risk that images have more transferability than text, and the training and experiment have been done on an imbalanced Twitter dataset. Despite some promising results [17], it could not be reliable since there is a lack of ground truth information about users where there is a possibility of publishing fake data about them and predicting unobserved users on user features training. Training on small datasets makes it challenging for CNN to learn and classify meaningful patterns, and CNN cannot process long text sequences.

Other researchers focus on the components of modality, a trigger, a target and a holder to annotate sentences [300] or study factuality scale and certainty levels like Lee et al. and Beretta et al.'s works [301] and [302], such as uncertain, somewhat certain, certain, and underspecified. The authors expressed these using generalized linear models (GLM), getting 74 per cent and 76 per cent F-scores on rumours resolving tweets and rumour checking, respectively. Value Directed Acyclic Graph (DAG) represents a set of values, leveraging the partial order among the claims to analyse claims and find the information source's credibility [302]. The main limitation Lee et al.'s work [301] reported is that it does not consider a variety of sparse lexical cues in addition to the problem of long sentences and dependencies. We have noticed that this system's accuracy depends on a graph representation of partial order of values from the world's knowledge, so if some are not available, the result will be imprecise. Other research works focus on event factuality and taking into consideration polarity (positive and negative) and the degree of certainty with its modality particle (fact, counterfactual, possibly fact, possibly counterfactual) [72]. In this study, linguistic analysis of the stories has been done to detect the factuality of a tweet, and the problem is that some of the linguistic features of tweets may be absent due to different criteria like location, language, the writing style and readership.

Blodgett et al. [303] discussed some essential linguistic features that contribute to factuality of quoted information in Twitter, distributed in five cue groups: report, belief, knowledge, doubt and perception [303].. The problem with this approach is that there is considerable differentiation in dialects where dialectal language identification is different according to the location, and it needs too much effort to deal with it. Rule-based approaches are used in Stanovsky et al. [304], Kilicoglu et al. [15], and they need more effort to form the rules manually. Stanovsky et al. [304] built a unified factuality dataset, where machine learning (support vector model) with the rule-based is implemented to solve the small dataset problem. One of the main limitations of this research is applying handcrafted dictionaries and rules where automatic training is more valuable [15]. Neural models for event factuality prediction have been implemented and reported in different chapters [7], [13], [14]. Generative Adversarial Network with Auxiliary Classification (AC-GAN) captures more syntactic information that helps check the veracity of events by considering shortest dependency paths as a central feature [13]. Enhanced Sequential Inference Model (ESIM) applies the BiLSTM CNN model to obtain a rich representation of statement pairs [14]. In [7], the authors developed a bidirectional child sum dependency tree LSTM (T-BiLSTM) and a bidirectional linear chain LSTM (L-BiLSTM). They discriminate between two categories of predication interaction: inside the context, information derived from the arguments of a predicate—for example, from determiners, like some and no, and outside context, is extensively studied in the domain of clause-embedding predicates, which fall into at least four distinct categories.: factives, like know and love.

Convolutional Neural Networks (CNNs) are used to extract local features from text and images [19], with a particular emphasis on unigram word features [168]. Combining image and text processing requires separate tasks and diverse datasets, and an additional risk is that the image and text are unrelated if the image is correct, and the text is correct. Multiple convolutional layers are used to combine the image and text input representations. This model achieves competitive results but requires a large amount of data to train; in [168], an early detection system for detecting false claims immediately after posting is implemented by training a merged CNN and RNN model. Additionally, Generative Adversarial Networks (GANs) have been used and extended to develop a "general feature set" for fake news across events to achieve early detection of false news [20].

To our knowledge, no approach in current fact checking models incorporates additional knowledge into evidence such as warrants. The majority of fact checking models limit their labels to sentences derived from the evidence. As discussed in chapter 5, we must consider how to address the challenge of insufficient evidence by leveraging warrant information that is currently ignored by current models. The development of such justifications warrants has received less attention in recent fact checking models, despite the warrant's potential for further improvement. As discussed in chapter 6, we propose new models for generating warrant to address this issue.

The majority of state-of-the-art models employ deep learning models, and due to the ambiguous nature of deep learning models that take on this prediction label, it is critical for the end user to understand how the model reasoned in order to arrive at the factuality prediction in order to persuade the user to trust the system. Despite the critical nature of explaining fact checking to end users, currently available solutions for explainability in the area of fact checking are lacking. The overall goal of chapter 7 is to generate factual claims that are more robustly supported by evidence that explains how the model arrived at a particular prediction.

3.4 Multi-task Approach for Joint Prediction of Rumour Stance and Veracity

The multi-task approach attempts to tackle stance classification and truth prediction concurrently rather than two separate tasks, as a result of their closeness. Some studies employ stance detection to improve performance and use the labels extracted from them as an input feature for veracity prediction models [255], [278], [279], [305]–[308]. These studies demonstrate that stance detection and the labels extracted from them are critical indicators for predicting the veracity of rumours [255], [278], [279], [305]–[308]. They combine the stance detection and rumour veracity classification tasks by utilising the concept of multi-task learning in a variety of ways, including parallel feature learning [96], [306]–[308], and hierarchically structured design [279]. Ma et al. [307] employ the GRU layer for each task, and the tasks also share a GRU layer to acquire patterns common to both tasks. As with Ma et al. [307], Wei et al. [279] used joint learning with a common layer and task-specific layers. Both models omit user data, whereas Li et al. [96] incorporate user credibility data in addition to the attention mechanism.

Tables 3.2 and 3.3 compare the performance of various methods for classifying rumour stances (single task) and veracity (multi-task) [279]. For stance and rumour detection, the macro-averaged F1 of the Hierarchical graph convolutional network GCN-RNN [279] and Hierarchical-predicting rumour Stance and Veracity [279] are superior to the baseline models evaluated [10], [306]. The post representation is obtained in Khandelwal and Peters & Cohan [285], [309] using a pre-trained Longformer and sliding window-based self-attention. The models reported in Enayet & El-Beltagy [305] and Li et al. [310] achieve competitive results in the SemEval 2017 and SemEval 2019 rumour detection tasks. Khandelwal [285] demonstrated that a multi-task approach for jointly predicting rumour stance and veracity using deep learning models such as BiLSTM outperforms previous rumour stance classification methods and veracity prediction on the SemEval 2019 Task 7 dataset. Macro-F is calculated by arithmetic mean (also known as unweighted mean) of all per-class F1 scores.

Table 3.2: RumorEval 2019 test results for Task A: Stance Detection.

System	**Macro-F**
Khandelwal's [285]	0.6720
Hierarchical graph convolutional network GCN-RNN [279]	0.540
Top-down tree structure using a recursive neural network TD-Recursive Neural Network (RvNN) RvNN [10]	0.509

Table 3.3: RumourEval 2019 test results for Task B: Veracity Prediction.

System	**Macro-F**
Li et al.'s model [96]	0.606
Khandelwal's [285]	0.5868
Hierarchical- predicting rumour Stance and Veracity PSV [279]	0.588
MTL2 (Veracity+Stance) [306]	0.558
BranchLSTM+NileTMRG [306]	0.539

Although these tasks are closely related and that multiple people's stances can be used to predict the claim's absolute veracity, state-of-the-art methods for false information detection are typically proposed for either stance detection veracity checking separately; stance aggregation features are required for effective veracity prediction [262]. According to the RumourEval 2019 report, the majority of systems are superior at detecting stances or predicting rumour veracity because they are trained independently without utilising multiple tasks simultaneously learned by a shared model, perhaps they are related, but not both. This constraint limits the generalizability of models. Additionally, as mentioned in the introduction section, previously published works were limited in their ability to verify the veracity of individual claims without considering all claims that addressed the same specific topic, which meant that many contradictory claims could be categorised as the same. In other words, because each claim expresses an attitude toward specific targets, one claim may be true toward one target but false toward another.

Based on the foregoing considerations, this work proposes combining the two tasks and learning them together to aid stance detection-based veracity prediction. Additionally, to differentiate fake from genuine information, the source's reliability indicates whether the source is more reliable, hateful, or biased than other information sources, all of which affect the veracity of published information. So, this work incorporates user reliability estimation.

As multi-task learning, which aims to perform stance detection and fact checking simultaneously, has gained popularity in the research field and continues to be heavily investigated. Motivated by a gap in which current multi-tasking models detect stances without encoding the specific target and current works still exhibit suboptimal performance, chapter 9 extended the work by proposing a novel framework for stance detection and veracity prediction that takes source credibility into account and compares the strength of arguments clustered by target in predicting the truth.

3.5 Truth Discovery

In contrast to fact checking, which compares each claim to a piece of evidence (unauthorised source like personal account), truth discovery examines contradictory information from multiple sources [311], [312]. In [313] Li et al. discuss several different strategies. In general, there are four types of methods that have been used in previous research to ascertain truth:

- Iterative methods where the trustworthiness of sources and the confidence of claims from each other are computed iteratively until convergence [314]
- The optimisation that measures the difference between the information provided by sources and the truth-based methods [315]
- Probabilistic graphical model-based methods where expectation maximisation is commonly used to infer the latent variables (parameters of truth and source reliability) [74]
- Neural networks [316]

Researchers have developed a variety of techniques for determining correct data from multiple sources of conflicting data, such as TruthFinder [317] and Voting [318] are tools for iteratively updating the reliability and accuracy of sources. Other works [319]–[323] employ additional factors to aid in truth discovery, such as information extraction techniques such as entity profiling [321] and knowledge graph [323]. Recently, truth discovery has been formulated as an optimization framework [324]–[326] that iteratively updates truths and source reliability.

Other works take a probabilistic approach, incorporating source reliability as a random variable into the probabilistic models and optimising the likelihood or posterior distributions of multi-source data. Wang et al. [248] developed a source reliability estimator. Samadi et al. [292] estimated source reliability and claim correctness using the Probabilistic Soft Logic (PSL) framework, whereas Nakashole & Mitchell [75] used language objectivity analysis in addition to Subject-Predicate-Object (SPO) triplets to determine the veracity of value. Other authors take a more practical approach to truth discovery; Wang et al. [327] divide sources and values into user-defined categories and then assess the information's credibility. Probabilistic graphical models can be used; Zhi et al. [311] used probabilistic graphical models with three measures to infer the truth value: silent, false spoken, and right spoken rates, while Zhao et al. [328] used a generative modelling process. Bayesian analysis can be used to determine source dependence [318]. Bayesian probabilistic modelling of the relationships between source quality, truth, and claimed values [329] estimates source reliability by taking into account the estimation's confidence interval [324].

Neural network models exhibit competitive accuracies in the truth discovery task [316], [330]–[332]. Despite demonstrating a significant performance improvement when using stance information in their rumour detection model, they rely on hand-crafted user features such as follower count and post count to reflect user credibility, which is separated from stance labels for predicting rumour veracity [96], [278], [305]. Enayet & El-Beltagy’s model [305] performed admirably in RumourEval 2019's stance classification system. Numerous rumours originate on either fake news websites or Hyperpartisan websites [333]. Liu and Wu [168] constructed user representations using network embedding and demonstrated the importance of user credibility information in determining the veracity of rumours. We consider the argumentation-based approach to discovering and justifying truths in our research in chapter 9.

3.6 Argument Generation

Numerous works have concentrated on identifying claims within the context of argument mining [334]. To accomplish this, the work in [335] demonstrates the critical nature of taking the conclusion and premises' primary objectives into account. Other studies use alternative methods for generating claims, such as opinion summarization, to capture the text's most salient points [336], [337]. While Egan et al.’s method [338] for summarising points made in online political debates relies on verbs and their syntactic arguments to identify silent information in political debates. From the premises, the conclusion (or claim) can be constructed, including its stance (for or against) the target [16], [339]. Hua & Wang [340] and Hua et al. [341] attempt to generate counterarguments to a given statement, whereas Wachsmuth et al. [342] and Hidey & McKeown [343] edited an original claim from the comments to generate new claims. Reisert et al. [159] summarise the text's main points using the Toulmin model and the relationships between the model's components and then use the summarised text to generate new claims automatically. In terms of the Toulmin argument model, Reisert et al. [159] construct complete arguments based on Toulmin's model [344], which requires that a claim be substantiated by data and justified by a warrant. They confine themselves to logical argument structure and grammatical rules to generate arguments about debate claims. Composing complex linguistic rules is challenging due to varying levels of knowledge about the language's syntactic structure and the requirement for extensive domain knowledge.

From a computational linguistics perspective, argument mining identifies and extracts the structure of inference and reasoning [140], [156], [157], [345]. Certain works, such as [160], [346]–[349], concentrate on argumentation-based reasoning using the Belief–Desire–Intention software model (BDI). The only one that is based on the Toulmin model of argumentation is Gabriel et al. [160], who use a multi-agent system to reason about uncertain beliefs and generate a new view based on available evidence while taking into account unclear and conflicting information. To generate a new belief based on the Toulmin model, data, justifications, and rebuttals, as well as argumentation-based reasoning, are developed in Belief–Desire–Intention software model (BDI) agents [160], [161]. Niven and Kao [174] demonstrate that spurious cues are critical in adversarial attacks and contribute to the generation of negated arguments. Additionally, they demonstrate BERT's capacity for learning.

Generating high-quality arguments is critical for factuality checking tasks, especially when dealing with contentious issues, as they are approached from a variety of perspectives, i.e., supporting arguments and counterarguments. Generating topic arguments enables the verification of an argument's (claim's) veracity based on available evidence and substantiated data. For example, the process of developing claims or reasons for taking a particular stance on a subject has been studied and analysed in numerous articles [336], [337], [340], [350]. Hua & Wang [333] and Wang & Ling [337], [340] used neural networks with a sequence-to-sequence mechanism to generate arguments and counterarguments. Other studies take a different approach, for example, summarising the main points of a debate [338] or recomposing existing text segments into new arguments by "recycling" topics and predicates [336]. El Baff et al.’s [350] work incorporates argumentative and rhetorical considerations to generate arguments, which has been demonstrated to be critical information for persuasive arguments [342]. Reisert et al. [159] use the Toulmin argument model to combine comprehensive arguments from a set of selected topic stance relations, where the claim is supported by data that is justified by a warrant. Similarly, to our work, those authors rely on a pool of argument components from which they construct arguments. According to Reisert et al. [95] and Toulmin [155], [159], the Toulmin model, which consists of six specified components, has a sizable influence on modern argumentation models: 1) a piece of data, 2) a subjective claim, 3) a warrant that logically connects the claim to the data, 4) the rationale for the claim, and 5) a degree of confidence (qualifiers) 6) a rebuttal to the claim [155], [351].

Argumentative text generation is used in a variety of works, for example [335], to investigate the question: to what degree can the conclusion of an argument be reconstructed from its premises? They devised two complementary strategies: one in which the top-ranked target is chosen as the conclusion target, and another in which a new conclusion is generated. Generating the conclusion of an argument from its premises requires three steps: (1) inferring the conclusion's target, (2) inferring its stance, and (3) phrasing the conclusion's actual text. Rather than viewing a claim as True or False, Chen et al. [3] propose viewing it from a variety of perspectives to gain a better understanding of it. They released a dataset containing claims, perspectives, and evidence, and the task is to identify the set of relevant argumentative sentences that represent perspectives for and against the claim. Given a debatable natural language claim, the system is expected to generate a diverse set of well-corroborated perspectives taking a position on it. Park et al. generate claims in response to a given claim, utilising a diversity penalty to encourage the presentation of diverse perspectives [339].

Multiple perspectives on a contentious issue are critical for avoiding bias and aiding in the formulation of rational decisions. We note that the current model constrains diversity. Diversity can aid in the development of more varied perspectives, thereby making perspectives appear more natural. This is because the conventional mechanism of attention is biased toward a single semantic aspect of the claim, whereas the claim may contain multiple semantic aspects. Additionally, ignoring common sense knowledge may result in the generation of perspectives that contradict well-established facts about the world. Thus, by taking these issues into account, chapter 8's work on argument generation could be significantly improved.

3.7 Warrant Generation

A few works have studied and analysed the task of generating the connection between the claim and the data. In our work, this is referred to as the warrant; in other works, it is referred to as the enthymeme [352] or implicit premise [353], which is typically the warrant (or major premise). Reisert et al. [159] assume that the data are accurate: If the data are accurate, the argument is true. The authors develop a model to generate Toulmin's argument using NLP techniques and some linguistic rules. They demonstrate that argument generation requires a greater understanding of language and complex reasoning and that their system requires significant development to perform argument generation. Boltuzic and Najder [354] investigate how to identify such implicit knowledge by analysing a large amount of text data from a variety of sources. In Habernal & Gurevych's work [354], the warrant is implicit because it is obvious from the statement's meaning, but Rajendran et al. indicated that if it is explicitly required, the argument synthesis method should be used [163]. Rajendran et al. [163] propose a method for creating a premise similar to a warrant in online review opinions that connects an aspect-related opinion to an overall opinion. However, their work's annotated dataset was insufficiently large to be useful for deep learning models. Singh et al. [355] manually generate a warrant in response to a claim and supporting evidence. In Horne & Adali's work [70], human workers are asked to think and write what they believe is necessary to explain why the provided evidence supports the provided claim.

Despite the critical role that warrants play in improving fact checking models, few studies have focused on it; also, the shortage of annotated warrant data motivates us to generate warrants, as shown in Chapter 6.

3.8 Datasets for False Information Detection

False information may spread and circulate through a variety of sources, including news agency websites, search engines, and social media. These sources were used to create a variety of datasets for training and testing new models for detecting false information. As shown in table 3.4, this section discusses several recent and widely used benchmark datasets for false information detection.

3.8.1. Perspectrum Dataset

Perspective is a neutral belief as a third-party partner to obtain different representations that emphasize vital content information and its sentiment fairly and accurately. Callison-Burch et al. and Yin et al. [3], [356] show that better decisions towards a claim could be made by creating different perspectives (viewpoints) and a better understanding of controversial issues. For example, to evaluate our proposed model, we compare our, while claim B has a refuted relationship with perspective B, perspectives are generated based on claim text. In other words, rewording claims have supported or undermined relations with perspectives that have supporting evidence.

Table 3.4: Datasets of false information detection

Dataset	**Description**
Kaggle [357]	This dataset includes nearly 13 000 posts from 244 distinct open sources labelled with fake, state, satire, conspiracy, hate, junk science and bias.
Baly et al. Dataset [358]	This dataset contains 1066 records which as used to detect bias in information and its veracity. Seven levels of bias are categorised ranging from extreme-left, left, centre-left, centre, centre-right, right to extreme-right. Then, veracity classes are categorised in one of three different labels: low, mixed, high.
NBCTwitter Propaganda Accounts Dataset [359]	The dataset contains information from NBC news focusing on users' tweet information, e.g., number of retweets, hashtags, the number of followers and mentions for the tweets.
MediaEval [360]	The dataset contains 7,898 false tweets and 6,026 true tweets.
RUMDECT [11]	498 events 494 events 2,313 events 2,351 event.
PHEME [361]	PHEME data consists of 1,972 false tweets and 3,830 true tweets and their replies for several real-world events.
Emergent [257]	It contains 2,595 pieces of news related to 300 events.
BuzzFeedWebis [36]	363 posts 1,264 posts.
LIAR [117]	12,836 pieces of news.
Media_Weibo [362]	40k tweets with images.
DeClare [363]	13,525 pieces of news.
FakeNewsNet [364]	211 pieces of news.
RUMOUREVAL[286]	325 source tweets related to 9 events
FactBank Dataset [275]	Factbank has 3864 sentences and 13506 event factuality values.
UDS-IH2 Dataset [7]	The largest event factuality dataset of UW, FactBank, and MEANTIME combination
FEVER	dataset of 125,000 claims
Perspectrum	about 1k claims, 8k perspectives and 8k evidence paragraphs
Argument Reasoning Comprehension Task ARCC	1,970 claims

Claim A: "A government must lessen the economic gap between its rich and poor citizens". Perspective A:" The Rich Poor Gap Silences the Political Voice of the Poor". Evidence A:" Research has also demonstrated a connection between economic inequality and political voice. The political process is far more responsive to the privilege's claims, and the privileged are better organized and engaged in the political process than are less affluent citizens. Recent studies show that government officials are far more likely to support the wealthy's policy preferences than those of the poor. In short, there is considerable evidence to suggest that there is a growing divide between those who have wealth and political influence and those who do not. Yasmin Dawood, THE NEW INEQUALITY: CONSTITUTIONAL DEMOCRACY AND THE PROBLEM OF WEALTH, Maryland Law Review: 2007".

Claim B: "Internet access is a human right".
Perspective B:" It is a big problem; too many people are file-sharing".
Evidence B:" The plan to slow down or stop internet connections is the most economical and practical way to deal with file-sharers. Many illegal downloaders are young people, and this plan will prevent the offenders from receiving a criminal record".

Yin et al. [3] build a dataset that helps for training and testing systems for the task of substantiated perspectives discovery given a claim and has a stance regarding it, supported by evidence texts, and annotated as in the following example in figure 3.2. In this figure, a claim is related to multiple perspectives with either support or opposing stance regarding a claim. Each perspective should have supported evidence to prove it. Yin et al. [3] confirmed that analysing diverse perspectives for a claim improves the ability to understand debatable claims.

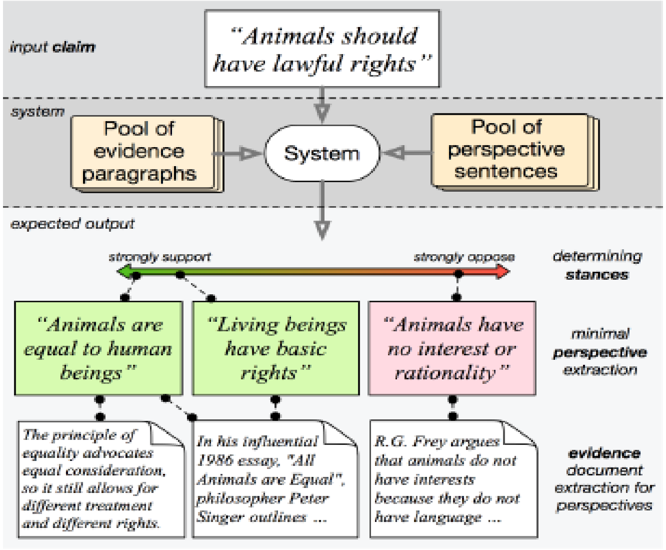

input claim
"Animals should have lawful rights"
system
Pool of evidence paragraphs
System
Pool of perspective sentences
expected output
strongly support
strongly oppose
determining stances
"Animals are equal to human beings"
"Living beings have basic rights"
"Animals have no interest or rationality"
minimal perspective extraction
The principle of equality advocates equal consideration, so it still allows for different treatment and different rights.
In his influential 1986 essay, "All Animals are Equal", philosopher Peter Singer outlines ...
R.G. Frey argues that animals do not have interests because they do not have language ...
evidence document extraction for perspectives

Figure 3.2: An example of a claim with its perspectives and evidence from PERSPECTRUM Dataset [3]

Table 3.5 shows the statistical information of the Perspectrum dataset.

Table 3.5: A summary of PERSPECTRUM statistics [3]

Split	**Supporting Pairs**	**Opposing Pairs**	**Total Pairs**
Train	3603	3404	7007
Dev	1051	1045	2096
Test	1471	1302	2773
Total	**6125**	**5751**	**11876**

In chapter 7, the Perspectrum Dataset is repurposed to produce factual statements, and in chapter 8, it is employed to generate perspectives.

3.8.2. Emergent Dataset

For contradictory claims, Gorrell et al. [286] developed the SemEval-2019 Task 7 dataset, and Ferreira and Vlachos [257] developed the Emergent dataset, which contains 300 claims and 2,595 associated news articles.

Additionally, chapter 9 evaluates the proposed framework's performance using an additional dataset, the Emergent corpus, because headline annotations direct readers' attention to the article, and Emergent is a dataset of rumours (claims) combined with news headlines and their stances. Because this model is focused on generating conclusions for news articles and condensing a lengthy article into a conclusion, it makes use of additional information, such as the headline in Emergent data, which represents the news store. Tables 3.6 and 3.7 illustrate the statistical data for the Emergent datasets.

Table 3.6: Emergent dataset [257]

Claims	300
Headlines	2,595
Minimum number of articles per claim	1
Maximum number of articles per claim	50
Training instances	2,071
Test instances	524

Table 3.7: Statistics of the Emergent dataset

Subject	Stance	Emergent Number	Percentage
Training	agree	992	24.37
	disagree	303	7.44
	discuss	776	19.06
	unrelated	2,000	49.13
		4,071	
Testing	Agree	246	24.02
	disagree	91	8.89
	discuss	776	19.06
	unrelated	500	48.83
		1,024	

3.8.3. Rumour Eval-2019

This study ,in chapter 9, applies the evaluation metric [324] to the data released at Rumour Eval-2019 for both stance detection and veracity prediction. It evaluates performance on both tasks using macro-averaged F1 because it overcomes the problem of imbalanced data. Tables 3.8 and 3.9 detail the statistical information for the Rumour Eval-2019 datasets.

Table 3.8: Rumour Eval-2019 Task A corpus [286]

	Support	Deny	Query	Comment	Total
Twitter Train Reddit Train	1004 23	415 45	464 51	3685 1015	5568 1134
Total Train	1027	460	515	4700	6702
Twitter Test Reddit Test	141 16	92 54	62 31	771 705	1066 806
Total Test	157	146	93	1476	1872
Total Task A	1184	606	608	6176	8574

Table 3.9: Rumour Eval-2019 Task B corpus [286]

	True	False	Unverified	Total
Twitter Train Reddit Train	145 9	74 24	106 7	325 40
Total Train	154	98	113	365
Twitter Test Reddit Test	22 9	30 10	4 6	56 25
Total Test	31	40	10	81
Total Task B	185	138	123	446

3.8.4. Argument Reasoning Comprehension Task (ARCC) Dataset

ARCC is a task with the responsibility of selecting the appropriate warrant from two reasonable candidates given a topic, a premise, and claim information [365], [366]. Habernal et al. [365] created a dataset for the Shared Task that contained 1,970 instances. It covers contentious news topics such as immigration and international affairs. This data set is used in chapters 5 and 6.

3.8.5. News Articles [1]

They created a corpus of documents from legitimate English news sources for the trusted category and unreliable news sources for the satire, hoaxes, and propaganda categories, using a dataset of 74K news articles collected from deemed websites. We will refer to this as a news corpus, in which the authors summarise the statistics and sources for each of the four classes in their respective partitions (train, dev, and test), as shown in tables 3.10 and 3.11. This data set is used in chapter 4 for our proposed domain adaption model.

Table 3.10: Statistics about the news corpus [115, pp. 1856-Table 5]

Source	**#Source**	**#Articles**	**Train**	**Dev**	**Test**	**Length(tokens)**
Trusted	4*	5,750	3,997	1,003	750	522±429.13
Satire	3	5,750	3,981	1,019	750	324±276.31
Hoax	2	5,750	4,014	986	750	262±300.92
Propaganda	2	5,330	3,670	910	750	1,047±1,156.87
Total	11	22,580	15,662	3,918	3,000	529±705.34

Table 3.11: Data Sources for news corpus [115, pp. 1856-Table 5]

Source	Trusted	Gigaword News*
	Satire	The Onion • The Borowitz Report • Clickhole
	Hoax	American News • D.C. Gazette
	Propaganda	The Natural News • Activist Report

3.8.6. Twitter

Volkova et al. [122] have built a dataset by collecting tweets from various accounts sources. The statistical information of the Twitter corpus is shown in table 3.12. Our proposed domain adaption model in chapter 4 is based on this data set.

Table 3.12: Twitter dataset statistics [122]

TYPE	NEWS	POSTS	RTPA	EXAMPLES
Propaganda	99	56,721	572	ActivistPost
Satire	9	3,156	351	ClickHole
Hoax	8	4,549	569	TheDcGazette
Clickbait	18	1,366	76	Chroniclesu
Verified	166	65,792	396	USATODAY

3.8 Summary

This chapter discussed the related studies on false information detection, which were classified into subtasks. Several limitations of current false information detection models are discussed in light of the information detailed in this chapter. This thesis suggests several limitations of current tasks for detecting false information based on a review of the literature. To begin, generalisation is limited due to a lack of labelled data, making it difficult to generalise and transfer existing models to new domains. Another significant limitation is that models perform poorly when they are based solely on evidence with insufficient information. Next, Scarcity of data and manual annotation are significant limitations, as humans are incapable of reading through vast numbers of social media posts and little work has been done to address the issue of scarcity labelled data. Finally, little research has been conducted on the topic of combining false information subtasks. The following chapters detail the procedures and experiments used to accomplish the thesis's objectives and address the thesis's research questions.

Chapter Four: **Linguistic Style-Aware Hybrid Model for Cross-Domain Factuality Checking**

4.1 Introduction

The previous chapters discussed the research questions, background on false information and methods to recognise false information. The primary goal of false information detection is to ascertain the truthfulness of a claim. This chapter addresses the research question concerning the detection of false information when no authoritative evidence is available. The research question of this chapter is RQ-1: “Can we improve the state of art performance for emerging claim verification? The reliability of information obtained via the Internet has emerged as a critical issue in contemporary society, profoundly affecting political and social affairs. Evaluating the reliability of various sources is a difficult task. Traditionally used fact checking models are based on labelled data or authoritative evidence, neither of which is always available. Due to the disparity in domain distributions between the source and target domains, these systems degrade when asked to perform outside domain test data.

Top-performing current state-of-the-art systems for automatic fake news detection are trained on vast quantities of labelled data and rely heavily on handcrafted feature engineering, as they are content-dependent and domain-specific. The primary disadvantages of the current models are their inability to address data scarcity and their poor performance when dealing with out-of-domain data. Several of the reasons for the performance degradation are due to the differences in the features of the source and target domains. We need to account for more robust and generalizable factuality checking across a broader range of domains.

In the absence of labelled data, it is well established that domain adaptation is an attractive option for training and testing models on various distributions. When trained on sufficient labelled data for specific domains, deep learning has demonstrated remarkable success at factuality control in various domains. Scraping data is a difficult task that requires considerable effort and is quite costly; to some extent, this is the domain of tasks like argument mining and related natural language processing tasks such as claim detection, premise detection, and so on. The adversarial domain adaptation model was proposed to utilise both labelled and unlabelled data from the source domain and related target domain data i.e., both the source and target domains share a common feature space but have distinct distributions [206].

To overcome these limitations, our model uses adversarial training-based domain adaptation to predict the factuality of text in an unsupervised target domain by leveraging a classifier learned from a supervised source domain. We evaluate our proposed models using datasets containing data from a variety of domains, and it outperforms the state-of-the-art system. Thus, it becomes more general and applicable to previously unseen data from a different domain.

The remainder of this chapter is structured in the following manner: the proposed deep learning-based approach is presented in section 4.2; we present experiments and results in section 4.3 that show results that suggest that the proposed model can generalize well to unseen data and conclude in section 4.4.

4.2 The Proposed Deep Learning-based Approach

We propose adversarial learning as a method for discovering shared and unique (domain-specific) characteristics between the source and target domains. In this work, we focus on adversarial domain adaptation for data forgery detection. This is accomplished through the use of labelled and unlabelled data from related domains. We propose a novel model that employs a gradient reversal technique to filter out (ignore) domain-specific features for adversarial learning-based domain adaptation, relying exclusively on shared domain invariant features.

The majority of the existing methods should include evidence to ascertain the extent to which trusted sources support or refute the claim or employ supervised learning methods. Due to the complexity of the tools and systems required to retrieve relevant documents, we focus on the way the claim is written, using text analysis to identify data, rather than retrieving evidence to verify the claim. Even though many studies focus on available data, we also attempt to learn a generalizable model from unseen or unlabelled data.

This chapter presents a factuality checking model that employs domain adaptation via adversarial learning [367] to select shared features from multiple domains rather than domain-specific features. Our model is composed of three modules: Feature Extraction, Representation, and Representation Extraction. These modules are used to extract semantic representations and features. The second module is style-based text classification, in which Multi-Layer Perceptron (MLP) and nonlinear activation functions are used to predict factuality labels such as trusted, satire, hoax, and propaganda. The final module is Domain Adaptation, which identifies the domain from which the features originated.

The general architecture of our model is depicted in Figure 4.1. We begin by pre-training a source encoder and classifier. Following that, we train the target encoder and discriminator in adversarial mode. To minimise the difference between the distributions of the target and source representations, the encoder maximises the domain classifier loss. In comparison, the domain classifier attempts to minimise it to extract common features across multiple domains.

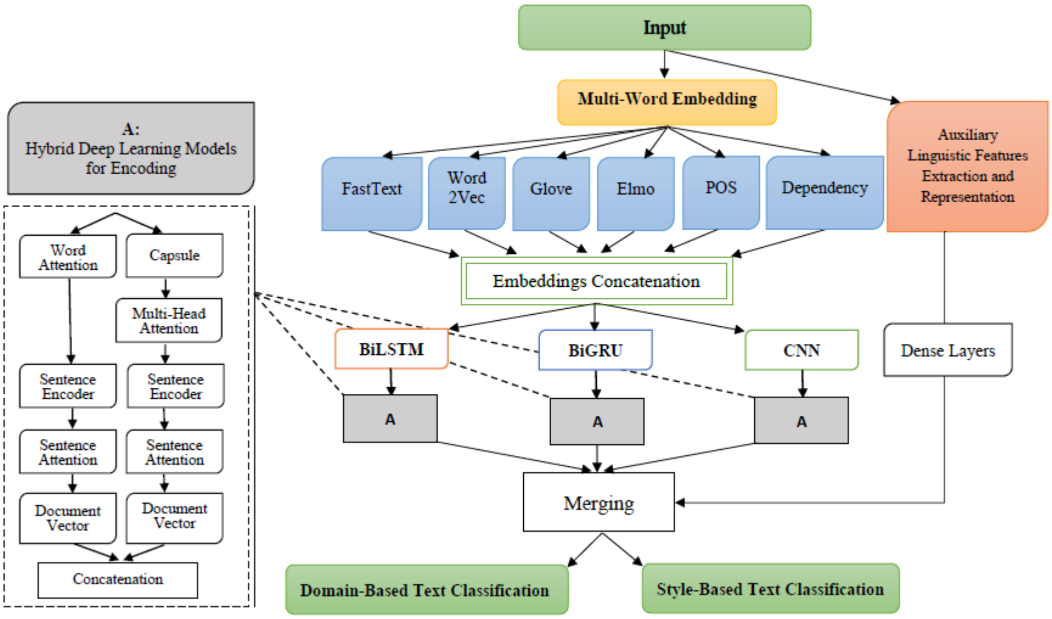


Figure 4.1: The general architecture of our model

We use multi-channels each of them with two-channel to obtain the final document vector, representing the document at a high level Due to the multi-channel nature of CNN, LSTM, and GRU, they perform better than CNN, LSTM, and GRU. The word sequence encoder, the word-level attention layer, the sentence encoder, and the sentence-level attention layer are among the deep learning components used. The model also includes a capsule layer, which employs the multi-head attention mechanism. To generate encoded sequences, we employ a variety of deep learning models, including Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Long Gated Recurrent Unit (BiGRU), and Convolutional Neural Network (CNN). Given that each deep learning model (for example, GRU, LSTM) has a distinct advantage, this situation demonstrates a highly efficient method for combining their benefits. The CNN model, which employs convolutional filters, captures local relationships between adjacent words in a sentence but not long-distance dependencies. On the other hand, LSTM overcomes CNN's short-term memory limitation by utilising inner cells. GRU is less complex than LSTM due to its fewer gates.

This work proposes a novel representation for documents to develop a more effective representation of documents and increase the precision with which semantic context vectors are generated. This is created by combining two fundamental attention techniques: Hierarchical Attention Generative Adversarial Networks and multi-head attention. Hierarchical attention weights can be computed at various levels, which, when processing large amounts of data, tends to concentrate on the distinguishing features, for example, the word-level and sentence-level, and can be used to aggregate significant words into sentences and then to aggregate significant sentences into a document. While a multi-headed attention process enables the model to focus on distinct positions from distinct representation subspaces. Concentrate on other vectors in the sequence to obtain a more accurate representation of this vector. When processing the vector, each attention head focuses on a distinct set of vectors.

4.2.1. Auxiliary Linguistic Features Extraction and Representation

This module takes input text and linguistic features to learn the semantic representation by implementing the proposed hybrid deep learning model. The linguistic (or auxiliary) features we rely on indicate the language writing style as in [123]–[125]. In addition to these features, we use other readability representations, the richness of vocabulary and News Landscape features (NELA) as they show noticeable improvement to evaluate the level of propaganda in Barrón-Cedeño et al.'s [115] work automatically. We apply three readability features: the Flesch–Kincaid grade level [368], the Flesch reading ease [368], and the Gunning fog index [369]. We consider five features of the vocabulary richness of an article, the type-token ratio (TTR), the number of types appearing exactly once or exactly twice in the article: the hapax legomena and dislegomena. As Barrón-Cedeño et al. [115, pp. 1854-Tables 2 & 3] summarize the most common vocabulary richness features and readability features.

The NEws LAndscape features (NELA) [115]: We consider NELA features,130 content-based NELA features such as sentiment, bias, and complexity, among others, are categorized in six subgroups, as in table 4.1.

We create an auxiliary feature vector using these features and then combine it with the encoder's output to generate enhanced text representation.

Table 4.1: NELA features [115, pp. 1854-Table 4].

Subgroup	**Description**
Structure	Part-of-speech (normalized counts)
Sentiment	Emotion: positive, negative, affect, etc. from Linguistic Inquiry and Word Count (LIWC), happiness score
Topic-specific	Biological process, relativity: motion, time, and space words, personal concerns: work, home, leisure, etc. (all from LIWC)
Complexity	SMOG readability measure, average word length, word count, cognitive process words from LIWC
Bias	Several bias lexicons, subjectivity probability in the text, A subjective example would be someone who believes pink is the best colour.
Morality	Features based on the Moral Foundation Theory [370], e.g. Being honest and trustworthy.

Multi-Word Embedding: Numerous embedding models are used as pre-trained input to capture various linguistic properties obtained by concatenating the corresponding word vectors from each model. Each word is transformed into a d-dimensional word vector in this layer, and the sentence representation is composed of vectors representing word embeddings. We use pre-trained FastText embeddings, WordVec [140], Glove embeddings [146], Elmo [147] for lexical embedding. For Syntactic Embedding, after converting each word to its corresponding POS tag, a low-dimensional vector is generated by POS tag embedding in addition to dependency tree embedding [371]. Equation 4.1 is used to convert each word to its vector representation of K pre-trained word embeddings:

Equation 4.1 $\mathrm{e}_{\mathrm{w}}^{\mathrm{concat}} = \mathrm{e}_{\mathrm{w}}^{1} \oplus \mathrm{e}_{\mathrm{w}}^{2} \oplus \ldots \oplus \mathrm{e}_{\mathrm{w}}^{\mathrm{K}}$

$\oplus$ is concatenation operator$\boldsymbol{e_w^i}$ is the word embedding vector of w_i in the ith pre-trained embedding. The final sentence S is as equation 4.2:

Equation 4.2 $S = (e_{W1}^{concat}, e_{W2}^{concat}, e_{W3}^{concat}, \ldots, e_{Wn}^{concat})$

4.2.2. Hierarchical Attention Generative Adversarial Networks

We use BiLSTM, CNN, and BiGRU to represent words incorporating contextual information given the concatenation embedded word vectors in equation 4.3. By reading each sentence in two directions, from beginning to end (forward) and from end to beginning (reverse), BiLSTM and BiGRU extract contextual information about the current word (backwards). The final encoded representation combines the Bidirectional hidden state representation and the Bidirectional hidden state representation. Equations 4.4 and 4.5 in the case of BiLSTM, 4.6 and 4.7 in the case of BiGRU, and 4.8 in the case of CNN.

Equation 4.3 $x_{in} = e_w^{concat} w_{in}, n \in [1, N]$

Equation 4.4 $\overrightarrow{h_{in}} = \overrightarrow{LSTM}(x_{in}), n \in [1, N]$

Equation 4.5 $\overleftarrow{h_{in}} = \overleftarrow{LSTM}(x_{in}), n \in [N, 1]$

Equation 4.6 $\overrightarrow{h_{in}} = \overrightarrow{GRU}(x_{in}), n \in [1, N]$

Equation 4.7 $\overleftarrow{h_{in}} = \overleftarrow{GRU}(x_{in}), n \in [N, 1]$

Equation 4.8 $h_{in} = CNN(x_{in}), n \in [1, N]$

We use the word level's attention mechanism to determine the weight of each word, as the words in the sentence have varying degrees of importance and contribute differently. To compute the sentence Vector. The word-level query vector, which is randomly initialised and collectively learned during the training process, and the hidden state representation are used to determine the word's importance in the text fact checking task. We use equations 4.9-4.11 to generate a sentence vector based on the importance of each word.

Equation 4.9 $u_{in} = \tanh(W_w h_{in} + b_w)$

Equation 4.10 $a_{in} = \frac{\exp(u_{in}^T u_w)}{\sum_n \exp(u_{in}^T u_w)}$

Equation 4.11 $S_i = \sum_n a_{in} h_{in}$

Sentence Encoder: For each sentence vector, we also use BiLSTM (equations 4.12 and 4.13), CNN (equations 4.14 and 4.15) and BiGRU (equations 4.16 and 4.17) to encode the sentences and, finally, the document representation.

Equation 4.12 $\overrightarrow{h_i} = \overrightarrow{LSTM}(s_i), i \in [1, L]$

Equation 4.13 $\overleftarrow{h_i} = \overleftarrow{LSTM}(s_i), n \in [L, 1]$

Equation 4.14 $h_i = CNN(s_i), i \in [1, L]$

Equation 4.15 $h_i = CNN(s_i), n \in [L, 1]$

Equation 4.16 $\overrightarrow{h_i} = \overrightarrow{GRU}(s_i), i \in [1, L]$

Equation 4.17 $\overleftarrow{\mathrm{h}_\imath} = \overleftarrow{\mathrm{GRU}}(\mathrm{s_i}), \mathrm{n} \in [\mathrm{L}, 1]$

Sentence Attention: As word contributes differently to the sentence representation, each sentence contributes differently to document representation, which is similar to word-level attention to measure the importance of each sentence for factuality checking task. Equations 4.18-4.20 measure the importance of the sentences.

Equation 4.18 $u_{in} = tanh(W_s h_i + b_s)$

Equation 4.19 $a_i = \frac{exp(u_i^{\mathrm{T}} u_s)}{\sum_i exp(u_i^{\mathrm{T}} u_s)}$

Equation 4.20 $v = \sum_i a_i h_i$

Where $\boldsymbol{u_s}$ is sentence-level context vector, $\boldsymbol{u_w}$ is word-level context vector.

Multi-Head Attention: According to Vaswani et al. [212], it is valuable to linearly project the Keys, Values and Queries h times with different learned linear projections to dk; dv and dq dimensions instead of executing single attention, as in equations 4.21 and 4.22:

Equation 4.21

$$\mathrm{MultiHead(Q; K; V) = [head_1; \ldots, head_h], where\ head_i = Attention\ (Q_i; K_i; V_i)}$$

Equation 4.22 $\mathrm{MultiHead(v_i, v_i, v_i) = Concat(head_1, \ldots, head_h)}$

The most useful words for the classifier are highlighted, as they contribute the most to the classifier. As demonstrated in a real-world news example, the term "Trump" and the phrase "presidential candidate" are highlighted as advantageous classification features. Additionally, as demonstrated in the following example, fake news is more subjective, with emotional overtones, which aids the classifier model in accurately detecting fake news by comprehending the language patterns captured in text.

Fake news: Coronavirus is prevented by drinking a hot beverage.
Real news: Trump has been a presidential candidate three times, in 2000, 2016, and 2020.

4.2.3. Style-Based Text Classification

A SoftMax layer receives the output of a fully connected hidden layer. The features extracted by all hierarchical attention networks, BiLSTM, BiGRU, and CNN, are combined as a unified vector input to the full connection layer and then fed to the classification module, which uses the SoftMax activation function to compute the probability distribution of the fake news category. A domain adaptation module receives the same input. Following the prediction of y and the loss, the cross-entropy loss will be calculated as in equation 4.23:

Equation 4.23 $L_C = -y_i \log \widehat{y_\imath} - (1 - y_i) \log\left(1 - \hat{y}_i\right)$

4.2.4. Domain-Based Text Classification

The final goal of this module is to achieve a low domain classifier loss, which indicates that the extracted features are shared between the source and target domains. The encoder attempts to deceive the discriminator by predicting the incorrect domain label, 1 for News Article and 2 for Twitter Corpus. This can be accomplished by incorporating a gradient reversal layer [246]. This layer multiplies the received gradient by a negative constant (the gradient reversal constant), reversing the direction of the backpropagation of the gradient. Gradient reversal ensures that the domain classifier's feature distributions are as indistinguishable as possible between the two domains' feature distributions [233]. The discriminator is MLP with a SoftMax layer to extract the output, as in equation 4.24:

Equation 4.24 $\mathrm{P} = \mathrm{softmax}\left(\tanh(\mathrm{WDd} + \mathrm{bD})\right)$ [233].

For domain discriminator optimization, we use the cross-entropy loss as the discrimination loss as in equation 4.25:

Equation 4.25 $\mathrm{L_D} = -\mathrm{d_i} \log \widehat{y_\imath} - (1 - \mathrm{d_i}) \log\left(1 - \hat{y}_\mathrm{i}\right)$ [233].

4.3. Experiments and Results

4.3.1. Datasets

We rely on two publicly available datasets of false information from two distinct but related distribution channels: social media [122] and news articles [1]. The datasets contain data from a variety of domains and are sufficient for testing domain adaptation. These datasets are found in chapter 3, sections 3.8.6, and they come from several domains to train our proposed model on the source domain and apply it for the target domain.

4.3.2. Baselines

Potthast et al. [36] demonstrate that that classifying fake news solely based on writing style comparisons is ineffective in general and demonstrates that real data cannot always be distinguished from fake data solely based on writing style comparisons. The closest comparable works to ours are Ghanem et al. [372], Rashkin et al. [1], and Volkova et al. [122]. They sought to distinguish genuine news from satire, hoaxes, and propaganda, as well as clickbait, as described in [122], [372]. They discover that using word n-grams results in a significant decrease in performance for unseen data; our proposed model addresses this issue.

Rashkin et al. [1] used a variety of lexicons and dictionaries to analyse the language of false information: propaganda, hoax, and satire: Wiktionary was used to determine the subjectivity, hedges, and degree of dramatisation, while LIWC was used to determine other features such as personal pronouns, swearing, and sexuality. Their analyses reveal that fake news contains many subjective terms, superlatives, modal adverbs, and personal pronouns. They used the LSTM model, which concatenates the output of the LSTM with the LIWC feature vectors before proceeding to the activation layer. They demonstrate that the LSTM model outperforms the Maximum Entropy (MaxEnt) and Naive Bayes models and that adding LIWC features improves performance.

Volkova et al. [122] analysed false information on Twitter to understand better, revealing that fake tweets contain significantly more harmful words, bias markers, hedges, and subjective terms. Compared to satires and hoaxes, they examine different types of information, uncovering more morals in propaganda and clickbait. Their model combines graph-based, signal words, and syntax characteristics with a macro-F1 performance of 71%.

For the Emotionally Infused Network (EIN) in Ghanem et al.'s LSTM layer [372] followed by attention and dense layers is used, with and without the emotional features branch, in addition to other suggested baselines such as bag-of-words with a support vector machine classifier (BOW-SVM), model of the mean of the document's word embeddings, and Logistic Regression classifier. Table 4.2 compares the findings of our proposed model to the state-of-the-art model (EIN), Ghanem et al. [372], which outperforms the baselines results of Rashkin et al. [1] and Volkova et al. [122].

4.3.3. Results

According to Ghanem et al. [372], their proposed EIN model, which is based on LSTMs and incorporates an emotional feature, achieves promising results: 79.43 per cent macro-F1 for the news dataset and 59.7 per cent for the Twitter corpus, respectively. Our proposed model outperforms Ghanem et al.'s best work [372], achieving 81.36 per cent macro-F1 and 69.54 per cent for the news articles dataset and Twitter corpus, respectively.

Table 4.2: News article and Twitter corpus

Model	News Article Corpus	Twitter Corpus
BOW+SVM [372]	70.70	57.45
W2V+LR [372]	69.78	36.43
LSTM [372]	72.26	55.41
EIN [372]	79.43	59.7
Our proposed model	**81.36**	**69.54**

On a benchmark dataset, the proposed method outperforms several baselines and produces results comparable to those obtained by previously proposed cross-domain fake news classification methods.

We can see that simply extracting domain-invariant information and taking into account linguistic style features significantly improves adaptation performance for the fake news classification task. Through adversarial training, enhanced performance is obtained by extracting discriminative features that most strongly support the final prediction.

4.4. Discussions

Our findings indicate that including additional features improves the model's performance compared to the model that does not incorporate extra-linguistic features. Trusted data is less subjective, more personal, and more objective, free of exaggeration or bias. Due to the variety of written styles used in Twitter accounts, news articles are easier to analyse than Twitter. The written style in news articles is more formal and precise. When three channels of Hierarchical Attention Networks models are combined with different deep learning models such as CNN and LSTM Capsule models with a hierarchy scheme and the attention mechanism, the source news articles data can enhance the performance of the target Twitter corpus through adversarial domain adaptation. Attention networks with hierarchical structures could be used to capture the text's hierarchical structure As a result, we incorporate it into our model. Each of the deep learning techniques used contributes significantly to the extraction of richer textual information: CNN extracts local features, attention mechanisms extract contextual information from text, and BiLSTM and BiGRU are robust at detecting long-term dependencies. Capsule networks assist adversarial networks in bridging the knowledge gap between source and target domains. Our hybrid model achieves promising results for cross-domain training on news articles and Twitter by maximising the domain discriminator loss through adversarial learning.

In general, we demonstrate that when used in conjunction with enriched linguistic features, our novel model classifier produced the best results, confirming our initial hypothesis (H1 and RQ1) and outperforming other approaches.

4.5. Summary

We propose a novel hybrid model for factuality checking that is based on a combination of stylistic and deep learning features. This task focuses on detecting genuine news, satire, hoax, or propaganda across multiple classes. Because multiple deep learning models can capture more high-level shared features that can be transferred across domains, our hybrid model outperforms comparable models, as shown in this study. By incorporating the capsule network, the model's performance is improved by focusing on relevant and non-relevant features that the max-pooling technique may miss and effectively detecting implied features. They require less training time than bi-LSTM and CNN networks. We discover that the additional features in the network improve the proposed hybrid model's accuracy. For long sequences, both BiLSTM and BiGRU are more robust in capturing global features over a long distance by encoding tokens from both left and right. Our proposed model outperforms all current systems.

In this chapter, we discuss how to use style analysis to detect false information when the claim does not include a reliable source of evidence, as is the case with new emergent claims, making the verification process difficult. Thus, the style of the written claim may reveal the writer's intent, acting as a stylistic idea to recognising deception and intent to deceive. Additionally, we require a generalizable model to address the issue of not having sufficient evidence to substantiate unseen emerging claims. In the following chapter, we will attempt to improve the performance of current models in the presence of insufficient reliable evidence and an ambiguous justification for supporting or attacking, which necessitates recognising the implicit link between a claim and a piece of evidence (i.e., warrant). In chapter 5, we will examine the effectiveness of using warrants automatically for fact checking tasks.

Chapter Five: **Warrant Aware Fact Checking**

5.1. Introduction

In the previous chapter, chapter 4, we discussed how linguistic style can aid in the detection of false information for emergent data where new claims may not yet be referenced in credible source resources. In this chapter, we will discuss the situation in which claims are substantiated by using knowledge bases (i.e., evidence), where it is possible to ground a claim to a reliable source, but the evidence does not contain enough information (i.e., warrant) assisted them in comprehending the relationship between a claim and an item of evidence. Typically, this chapter aims to answer the research question RQ-2 "To what extent can external knowledge, such as a warrant, aid in fact checking performance improvement?".

An argument is composed of two critical components: a claim and a relevant piece of evidence. Fact checking entails making predictions about the relationships between these components. While it is possible to determine the truthfulness of claims by examining relevant evidence, establishing the relationship between the claim and the evidence remains a challenge. The most probable factuality label can be detected if a model can establish a connection, such as a warrant, between the claim and the piece of evidence. The claim and the evidence are used by the majority of recent fact checking DNN methods, but not the warrant. In some instances, the label was not discernible due to a lack of explicit reason from the evidence supporting the claim. As a result, a warrant is required to ascertain the veracity of the claim. The effectiveness of leveraging warrants for fact checking is therefore investigated in this chapter. The warrant is the logical inference statement that acts as a link between the claim and the evidence to extract supportive sentences; it is based on Hashimoto et al.'s proposed excitatory and inhibitory relations [373].Table 5.1 illustrates an example claim-evidence pair with two opposing candidate warrants, W0 and W1, for fact checking purposes.

Table 5.1: An example claim-evidence pair for fact checking

Evidence: Miss America gives honours and education scholarships.
Claim: Miss America is good for women
W0: scholarships would take women away from the home.
W1: scholarships would give women a chance to study.

The relation (i.e., the factuality of the claim) is decided based on other information; if the warrant is W0, then the relation with this is refuted while W1 gives support relation.

This chapter proposes a method for fact checking that makes use of warrant to improve the effectiveness of current false information detection. The proposed method is based on a publicly available dataset ARCC, which stands for Argument Reasoning Comprehension Corpus from News Comments [187], which was built for the 2018 SemEval task [366] by Habernal et al. [365]. We discovered that external information affects fact checking via deep learning. Thus, the algorithm minimises incorrect predictions by elucidating the rationale for its labelling decisions. This work demonstrates the possibility of further performance improvements through the addition of Toulmin model components i.e., warrant.

The rest of the chapter is organized as follows. In section 5.2, we propose a new model for fact checking model leveraging warrant. In section 5.3, the proposed fact-predictor architecture is presented. Experiments are presented in section 5.4 and we present the summary in section 5.5.

5.2. Warrant Aware Fact Checking

The closest work to our task is by Reisert et al. [159], which proposes a computational approach to generate Toulmin's argument using natural language processing techniques and some linguistic rules, where a greater understanding of language and complex reasoning is required. According to the authors, their work could be significantly improved for the task of argument generation.

To our knowledge, this is the first work to incorporate warrants for performance enhancements in fact checking. The closest work is Singh et al. [164] work although it is for evidence detection. Singh et al. [164] predict the relationship between a claim and its corresponding premise using a warrant to rank the evidence of a group of candidates and choose the best piece of evidence for a particular claim. They created a new dataset by randomly selecting correct warrants to create positive instances labelled with (Premise, Claim, and Correct warrant) and labelled with (1), as well as negative instances labelled with (0) for (Random Premise, Claim, and Correct warrant), in which they randomly select a premise for a claim with its correct warrant. In their model, warrants are ranked to determine the best rationale for the argument, obviating the need to search for a relationship between claim and evidence. The opposing claim's warrant is viewed as a rebuttal in the Toulmin argument.

They automatically extracted warrants for evidence detection from an existing, structured corpus of arguments ARCC considering the consistency between the given argument from Context-Dependent Evidence Detection (CDED) dataset [374]. They then used these warrants to augment the CDED evidence detection system. They discovered that, on average, the automatically acquired warrants are not high-quality. This can be attributed due to the very low shared lexical content between the two different datasets contained in this study, as in table 5.2.

Table 5.2: Performance of evidence detection in Singh et al.'s models [164], the state of the art model for using warrants for stance detection, they only use correct warrant

Model	Accuracy
Bidirectional LSTM model without a warrant	72.71
Bidirectional LSTM model with the correct warrant	76.74

We argue that a model should be used to determine the single warrant that best bridges the reasoning gap from a set of candidate warrants. The proposed best warrant selection model is only related to the work of Singh et al. [355]. The distinction between this work's best warrant model and theirs is that their model relies on crowd workers to select the most relevant and best warrants, whereas this work automated the ranking models rather than relying on manual methods. They used the IBM Context-Dependent Evidence Detection (CDED) dataset [374]. The following is a sample of the claim, evidence, and five candidates from Singh et al.'s ranking warrant dataset [355].

An instance of Toulmin argument example with a given claim and evidence pair and five candidate warrants (W1-W5), the ranks of the warrants are 3, 4, 2, 1, 5, where W3 can be considered better reasoning from evidence to claim.

- The claim: " We need libraries ".
- The evidence: " Libraries provides books and internet access to those who cannot afford it"
- Warrant 1: "Libraries are the most convenient medium to gain knowledge for the poors "
- Warrant 2: " Low income individuals deserve to have access to books and internet like their privileged peers do "
- Warrant 3: " Library that is accessible by books, internet access in free cost "

- Warrant 4: " A lot do not have access to books and computers to help live them or learn for a better life "
- Warrant 5: " Libraries are a societal necessity because they provide free access to books and the internet, and society benefits from informed and educated people"

Our proposed model looks for the best warrant from a given set of candidate warrants, that is related to a claim and then identifies the relationship between the claim and evidence i.e., support (if warranted), attack (if contradictory warranted), or irrelevant (if the warrant was chosen at random and is unrelated to the claim).

5.3. The Proposed Fact-Predictor Architecture

5.3.1. Key Idea

The proposed model's goal is to categorise the relative strength of the evidence supporting a claim. We discuss the importance of warrants as the foundation for logical inferences in determining claims-to-evidence relationships as in figure 5.1. To improve fact checking, this model incorporates the warrant and uses multi-channel combined with multi-head attention for fact prediction.

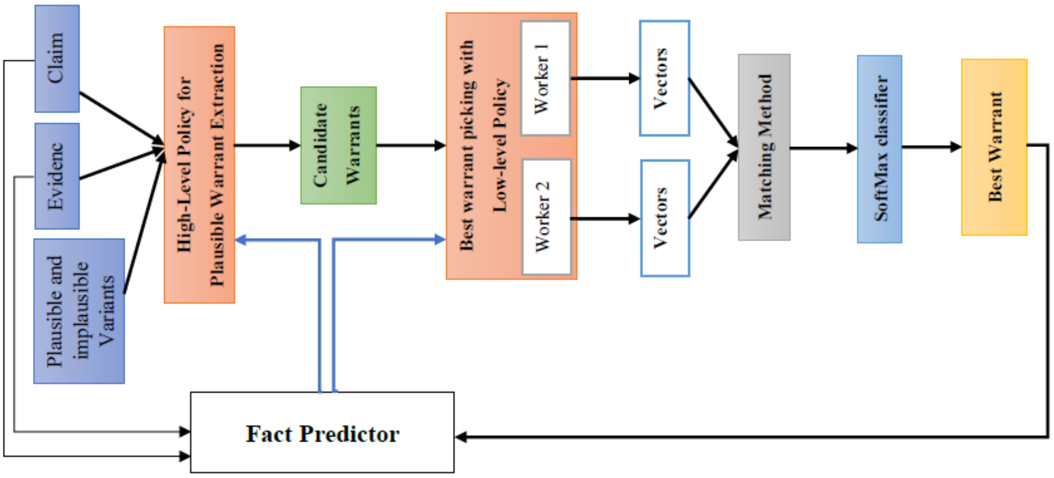

Claim
Evidenc
Plausible and implausible Variants
High-Level Policy for Plausible Warrant Extraction
Candidate Warrants
Best warrant picking with Low-level Policy
Worker 1
Worker 2
Vectors
Vectors
Matching Method
SoftMax classifier
Best Warrant
Fact Predictor

Figure 5.1: The proposed model architecture

The proposed model architecture is divided into three components: the first is a high-level policy for plausible warrants, and the second is a low-level policy for best warrant selection. The fact predictor is the third component, which is also used to reward behaviour that leads to the desired outcome.

The initial input is as triple (claim, plausible and implausible warrant, and evidence), as (c, w, e), and the input for fact predictor is a triple <c, w, e> where w is the best warrant. Hierarchical Reinforcement Learning HRL is used to select the best warrant for fact predictor, in which the high-level policy determines whether a warrant is plausible in light of the claim evidence pair, and the low-level policy is trained to select the best warrant. The low-level policy combines the outputs of knowledge-based and style-based models. Capsule and BiGRU networks are used to improve the representation of syntactic and semantic information in knowledge-based prediction for reasons that Bi-GRU is capable of reading text in both directions and extracting contextual, semantic, and grammatical information about the words in the text [375] and Capsule networks address the issue of information loss associated with CNN pooling operations by representing various attributes of a particular type of entity, such as an object or an object part results in higher classification accuracies [219].

To perform style-based prediction, we apply a feature-guided conditioned cycle GAN via VAE to convert the style of a text to the desired style and then check for style matching. Our work is predicated on the premise that compatibility between these two texts in the style space indicates greater consistency, i.e., that greater compatibility increases the likelihood that the warrant is selected. The final component of our model is the fact-predictor, which is used to determine the claim's veracity. Additionally, it serves as a guide for determining the relationship between a claim and the evidence, as well as for directing all policies.

The formulation of goal-directed lower-level policies is emphasised in hierarchical reinforcement learning approaches. We feed the higher-level policy's goals to the lower-level policy so that it learns to behave differently depending on which goals it is attempting to achieve. As a result, we formulated a goal-conditioned reward function to aid the lower-level policy's learning process.

5.3.2. A High-Level Policy for Plausible Warrant Extraction

The primary objective of the high-level policy is to distinguish relevant warrants from those that are irrelevant to the claim evidence pair. The relevant warrant is forwarded to the lowest level of policymaking to obtain the most appropriate warrant decision. The primary objective of warrant extraction is to ascertain whether or not there is a connection between the claim and the evidence supporting or refuting it. Relevant warrants should be more persuasive and relevant to be considered reasonable. The ARCC data is annotated with a topic t, a claim c, a set of warrants w (which connect the claim to the evidence) including plausible and implausible ones, and a piece of evidence e. In our work, we train our model to select the most appropriate warrant given a claim and evidence argument.

Policy: This policy depends on good text representations after the embedding vector represents each word with an attention mechanism. For each time step, each claim-evidence input pair, multiple warrants are examined for each claim evidence pair. Next, a deep neural network is applied: Bidirectional Long Gated Recurrent Unit (BiGRU) combined with the attention mechanism to capture crucial information from both directions. To encode the inputs and obtain the contextual information, the model uses BiGRU to efficiently use past features and future features and summarise these words information from both directions (forward and backwards) as in the equations 5.1-5.3:

Equation 5.1 $\overrightarrow{\mathrm{h}}_{\mathrm{i}} = \overrightarrow{\mathrm{GRU}}_{(\mathrm{c}_{\mathrm{i}})}; \quad \mathrm{i} \in [1,\ \mathrm{C}]$

Equation 5.2 $\overleftarrow{\mathrm{h}}_{\mathrm{i}} = \overleftarrow{\mathrm{GRU}}_{(\mathrm{c}_{\mathrm{i}})}; \quad \mathrm{i} \in [\mathrm{C}, 1]$

Equation 5.3 $\mathrm{h}_{\mathrm{i}} = \overrightarrow{\mathrm{h}}_{\mathrm{i}} \oplus \overleftarrow{\mathrm{h}}_{\mathrm{i}}$

Equations 5.4-5.6 are used to determine the attention weight a_i, between each claim's hidden state h_i , and the warrant representation, i.e., the similarity between the claim's hidden states and the warrant representation.

Equation 5.4 $\mathrm{h}_{\mathrm{w}}^{\mathrm{p}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{h}_{\mathrm{w}}^{\mathrm{i}} / \mathrm{n}$

Equation 5.5 $\mathrm{m}_{\mathrm{i}} = \tanh\left(\mathrm{W}_{\mathrm{c}} \cdot \left[\mathrm{h}_{\mathrm{i}}; \mathrm{h}_{\mathrm{w}}^{\mathrm{p}}\right] + \mathrm{b}_{\mathrm{c}}\right)$

Equation 5.6 $\mathrm{a}_{\mathrm{i}} = \mathrm{softmax}(\mathrm{m}_{\mathrm{i}}) = \frac{\exp(\mathrm{m}_{\mathrm{i}})}{\sum_{\mathrm{t}=1}^{\mathrm{C}} \exp(\mathrm{m}_{\mathrm{t}})}$

where h_w^p is pooling vector for the warrant, $\mathrm{h}_{\mathrm{w}}^{\mathrm{i}}$ is……, n is….m_{i} is…

The claim representation c can be derived based on the attention vectors a_i by equation 5.7:

Equation 5.7 $\mathrm{cr} = \sum_{\mathrm{i}=1}^{\mathrm{C}} \alpha_{\mathrm{i}} \cdot \mathrm{h}_{\mathrm{i}}$

The above equations are used to represent evidence by replacing claim c with evidence e. We propose a method that ranks multiple warrants for a given claim and evidence pair. It first feeds both claim and evidence representation to a SoftMax classifier where the highest probability stands for the best warrant to fill the gap between the claim and the evidence. Where $\oplus$ represents a connection of vectors, claim representation and evidence representation, equation 5.8:

Equation 5.8 $\mathrm{score} = \mathrm{softmax}(\mathrm{w}.[\mathrm{cr} \oplus \mathrm{er}] + \mathrm{b})$

It serves as a guide for selecting candidate warrants [w1;w2..wn] which support the claim and evidence with plausible reasoning.

The high-level policy utilises a reward to select the warrants to guide warrant extraction over the warrants sequence. The actions taken by this policy depends on the score result, which is a conditional probability of binary classification while the state at each time claims evidence pair. More details about the state, action and reward of the high-level policy are described as follows:

- ***State:*** the state is composed of three parts, the claim and its evidence in addition to a candidate warrant from the dataset. The policy uses this information to decide either to select the candidate warrant as relevant or not.
- ***Action:*** *the policy* samples *action* $ai,\ j\ \in \{0,\ 1\}$ by the conditional probability, as defined in score eq. 5.8, The actions are the high-level goals (the candidate warrants that should be fed to the low-level policy)
- ***Reward:*** After the policy has taken an action, This action should be rewarded with a cosine similarity between the vectors of the selected warrant and the claim evidence pair.

5.3.3. Best Warrant Picking with Low-level Policy

The high-level policy provides a candidate (plausible) warrant sequence w1, w2,..., wn; the low-level policy l selects the strongest (best) warrant from that sequence and discards the less likely ones. The policy has two workers which are trained independently. They each help the low-level policy learns to select the best warrant by considering different perspectives, i.e., worker one uses the semantic representation, and worker two the style representation. The low-level policy sends the outputs from both workers to the SoftMax output layer. For each warrant, the conditional probability by SoftMax is updated until all warrants are processed, and the best warrant is returned.

The states and rules for a reward for the low-level policy are as follows:

States: are the information of relevant warrant sequences that comes from the high-level policy, also, to claim evidence pair for deciding to select the best warrant.

Action: this policy adopts the SoftMax function to decide the best warrant based on conditional probability results, Thus, the action will either consider this warrant to be the best or will ignore it.

Reward: After the policy has taken an action, this action should be rewarded entailment metric between the hidden states of the selected warrant and the word embedding of the claim evidence pair.

Worker 1 Knowledge-based prediction (content-based representation): This worker's objective is to encode the input's syntactic and semantic properties. semantic representation of the input is based on the content of the text e.g., term vectors. Its input consists of a low-level representation of data (for example, claim words), and its output consists of an implicit representation of this data

In this work, a capsule network model incorporating BiGRU is proposed, BiGRU capsule networks. This model consists of two parts, the BiGRU module is used to capture input context features and the capsule module is used to obtain the spatial position relationships of local features. Initially, a claim is represented by a word embedding where words from the vocabulary are mapped to vectors. In this model, the pre-trained word vector GloVe is employed, it performs better and faster on our used dataset as equation 5.9:

Equation 5.9 $x_{in} = e_w^c w_{in}, n \in [1, N]$

BiGRU is used to capture long-distance dependencies within a sentence and proved its effectiveness to encode sentence representation. It captures the information from both directions left and the right context; then, the word representation is the concatenation for them, equation 5.10-5.12:

Equation 5.10 $\overrightarrow{h_{in}} = \overrightarrow{GRU}(x_{in}), n \in [1, N]$

Equation 5.11 $\overleftarrow{h_{in}} = \overleftarrow{GRU}(x_{in}), n \in [N, 1]$

Equation 5.12 $h_{in} = \overrightarrow{h_{in}} + \overleftarrow{h_{in}}, n \in [1, N]$

Capsule Network [213], [214], have generated Capsule model instead of the single neural node as in CNN models; they have used neuron vector for input and output layer with dynamic routing algorithm instead of pooling operations. It is used for understanding the spatial information and the contextual information of text in different tasks. For example, question answering [215], word segmentation [216] and extract the global semantic features of different categories [217], sentiment analysis [218]–[220], cross-domain [220], [221], sarcasm detection [222], propaganda detection [223].

According to Gao et al. [184] capsule network is robust to extract a richer representation of a text and other significant features such as word position the semantic and syntactic structure. This proposed model applies Gao et al. [184] equations to obtain the capsule network's output and considers other output of BiGRU. The outputs of capsules networks are achieved as equation 5.13-5.15:

Equation 5.13 $\hat{u}_{o|i} = w_{io} h_{in}$

Equation 5.14 $S_{out} = \sum_{i=1}^{m} c_{io}\, \hat{u}_{o|i}$

Equation 5.15 $c_{io} = \frac{exp(b_{io})}{\sum_k exp(b_{ik})}$

Where c_{io} is the coupling coefficient, is determined by the dynamic routing method and S_{out} is vector representation. The activation of capsule network output is calculated by the nonlinear function (Squash) for normalisation purposes, as shown in equation 5.16:

Equation 5.16 $v_{out} = \frac{\|S_{out}\|^2}{1+\|S_{out}\|^2} \frac{S_{out}}{\|S_{out}\|}$

Where v_{out} is the output vector of the capsule network. For c is v_{outc}, for w v_{outw} and for e v_{oute}. The dynamic routing method [220] is shown below:

```
for all capsule i in layer l and j in l+1:
initial: $b_{ij} \leftarrow 0$
for iterations do
```

Equation 5.17 $c_{ij} \leftarrow soft\,max(b_y)$

Equation 5.18 $s_j \leftarrow \sum_i c_{ij} u_{j|i}$

Equation 5.19 $v_j \leftarrow squash(s_j)$

```
For all capsule i in layer l and j in l+1: $b_{ij} \leftarrow b_{ij} + u_{j|i} \cdot v_j$
return $v_j$
```

After generating the vector representation for each claim, warrant and evidence, All the resulting vectors are concatenated and fed to a SoftMax classifier to predict the relationship between a claim and a piece of evidence (to express labels 0,1,2).

Worker 2 Style based prediction (semantic transformation): Stylistic feature-based representations are completely distinct from textual (content) representations and are capable of capturing significant characteristics of the text's writing style. A semantic transformation model is used to transfer the text semantic domain and then conduct the matching between the original text and the transferred text. The motivation behind using this style-based model was to match the target text's style with its original style and compare them.

Our model makes use of a CycleGAN network to power our style-based machine learning model. CycleGAN combines two generators and two discriminators, resulting in two bidirectional data input mappings; it also learned a transformation between image domains [376], [377]. The success of CycleGAN in image transformation and semantic matching [378], motivated us to propose an argumentative relation identification task based on style transfer. It makes use of the generative variational autoencoder, the CycleGAN architecture, and the generative variational autoencoder [379], [380].

This model has three texts, c, w, e. First, it checks the style of claim toward the warrant, c and w that respectively, transfer claim c to warrant style Qw and warrant w to claim style Qc., then the model matches each transferred text to the original one: c with Qw and w with Qc via Manhattan distance as it is accurate in determining the distance between real-valued features [381]. The same thing of evidence, only replace c with e. The model averages all Manhattan outputs the maximum average.

For both claim and evidence, warrant information is used to transfer text into warrant style and vice versa. To achieve that, the VAE generator network combines the same z from the claim and c from the warrant to generate various text that satisfies the new constraints encoded in a specific style and preserving the knowledge. A discriminator consisting of CNN estimates the probability that the transferred text comes from the target semantic style domain and determine how the generated text is acceptable. Generators and discriminators are trained using backpropagation. Next, we identify the most appropriate conditional statement from our model based on our semantic relation.

5.3.4. The Policy: The Hybrid Model of Semantic Transformation and Representation.

Detection is accomplished by merging vector representations from both style models and knowledge models, using product, concatenation, and difference matching methods. The outputs from all the subjects are concatenated, and a SoftMax classifier is used as in table 5.3.

Table 5.3: Vector’s representations of matching methods

Matching method	**The style-based vectors**	**The knowledge-based vectors**
Vectors Concatenation	(c+qc)(w+qw) (e+qe) (w1+qw1)	$(v_{outc}+v_{outw}+v_{oute})$
Vectors Elementwise product	(c*qc) (w*qw) (e*qe) (w1*qw1)	$(v_{outc}*v_{outw}*v_{oute})$
elementwise difference	(c-qc) (w-qw) (e-qe) (w1-qw1)	$(v_{outc}-v_{outw}-v_{oute})$

5.3.5. Fact predictor: Multi-Channel Multi-Head Attention Based BiGRU Siamese Network.

Word Embedding Layer: All inputs (claim, warrant, evidence) are fed to the input layer. Each input is connected to the embedding layer, which builds word embeddings using Elmo, GloVe and fastText. In this work, to build word embeddings, Elmo, GloVe and fastText generate a word vector table. For each input, all word vectors of the word embeddings generated by Elmo, GloVe and fastText are concatenated as a matrix that is finally fed as inputs C, W and E claim, warrant, and evidence to the BiGRU layer.

Word Encoder Layer: Each word in each input C, W, and E represents a multi-channel of word embedding. The Word encoder layer creates a new representation for each word by summarising contextual information from forward and backward directions using BGRU from both directions in a comment. For the whole input of (C, W, or E) to obtain hidden state representation *ht* for each word, forward hidden state and backward hidden state are concatenated for each word, and all of them are represented as H for each input.

Multiple-head Attention layer: For claim fact detection, each part in each input, claim, warrant, and evidence has a specific part with a variant role from different factors, so this model focuses on them by applying multiple heads of attention representing the semantics of the three inputs. After all hidden states have been fed to the attention layer as equation 5.20, each input's entire semantic representation is represented as equation 5.21, and equation 5.22 where Wk1 and Wk2 are weight parameters. The final input is C. Other feature vectors are merged to the final input representation, Linguistic features F: the sentiment feature vector and other Linguistic Inquiry and Word Count (LIWC) [128] features such as subjective, number, Swear, Negation and speculation expressions. Cnew is a new representation that combines the input representation with linguistic features, making the sentence more meaningful. The model generates a new representation V that passes through the SoftMax layer to determine the fact checking output label. 13 the prediction with the highest probability is the predicted fact. The best final application label is used as a reward for good HRL.

Equation 5.20 $A = \tanh(Wk1HT)$

Equation 5.21 $B = \mathrm{SoftMax}(Wk2A)$

Equation 5.22 $C = BH$

Equation 5.23 $Cnew = C + F$

Equation 5.24 $Label = \mathrm{SoftMax}(Wv\,V + bV)$

5.4. Experiments

We address the research question RQ-2 in our experiments by comparing the performance of a fact checking model with and without warrants.

5.4.1. Dataset

For warrant selection, the corpus of ranked warrants from Singh et al. [355] is used because it is the only dataset available for ranking warrants, where only the top-ranked warrants are kept. This dataset is annotated for warrant preference learning, where a list of warrants are ranked according to how well they connect a particular claim with a given piece of evidence. They collated all the warrants for 100 claim-evidence pairs, each pair being annotated manually with five warrants from the top (high score) to bottom (low score) ranked.

For the fact checking task, ARCC data is used, standing for Argument Reasoning Comprehension Corpus from news comments [187] that is the build for SemEval task 2018 [366] by Habernal et al. [365]. The argument reasoning comprehension task is to pick the right implicit warrant from two choices provided with an argument: a claim and a premise. For the evidence detection task, Singh et al. [164] modify this data to be more appropriate to decide the relation between claim and evidence. The relation label is either support or non-relevant. They label the datasets given the tuples of (Premise, Claim and Correct warrant) as a positive label, e.g., 1, and the tuples of (random Premise, Claim and Correct warrant) as negative label 0 (non-supporting).

Our proposed model's goal is to determine if the evidence supports, undermines, or is irrelevant to a claim, so it considers both the correct warrants that explain why a premise supports the claim and the alternative warrants that lead to contradictory (rebuttal) claims. The random warrant is no longer relevant information. The modified ARCC data is shown below:

- {claim, correct warrant, correct premise, label 1}, where 1 denotes verifiable fact: True
- {claim, random warrant(irrelevant), correct premise, label 0}, where 0 denotes irrelevant warrant: unrelated
- {claim, attack warrant(rebuttal), correct premise, label -1} where -1 denotes unverifiable fact: false.

5.4.2. Settings

The embedding matrix is utilised with word2vec embeddings. Models were implemented using Keras and TensorFlow. The proposed model used 20% of the data as test data. The list of hyperparameters used to train neural architectures is presented in table 5.4.

Table 5.4: Hyperparameters used to train the neural architectures

Hyperparameter	**Value**
Batch size	32
Embedding size	300
GRU cell size	128
GRU dropout	0.2
Optimiser	Adam
Learning rate	0.001
The number of route iterations	3
Regularisation constant of the dropout layer	0.2
The number of capsules	400

5.4.3. Results and Discussions

For warrant selection, results are evaluated by the normalised version of Mean Reciprocal Rank, Mean Quantile (MQ) score [382], which measures the correct ranks among all candidate warrants. We obtain an MQ score of 0.73 where the quantile ranges from 0 to 1.

More reasonably selected warrants increase the performance of this model. It is observed that the proposed model sometimes mislabelled the relation when the warrant has noise information or is less relevant. There is no comparison between the proposed fact predictor model with other works in experiments since no previous work considering the warrant and rebuttal for fact checking has been applied. The performance is evaluated using Accuracy, which is calculated as equation 5.25:

Equation 5.25 $Acc\mathrm{uracy} = \mathrm{T} / \mathrm{C}$

T is the number of correctly classified labels, and C is the total number of labels. The modified data has 1,210 instances as training data and 444 instances as test data for each relation, i.e., the model collects only correct warrants for support relation, alternative warrant (rebuttal) for attack and randomly warrant for no-relevant relation.

The experimental findings given in table 5.5 demonstrate that choosing the best warrant from a set of correct warrants, instead of a randomly selected correct warrant, increases the model's performance by providing more confidence with the designation. In the training data, the best test accuracy was 81.69 %, which was obtained by feeding the fact-checker the labelled first-ranked warrant. It is observed that the sentiment, negation and other style information help clarify the relationship to be captured. For example, if the warrant (or alternative warrant) has the same polarity with the claim and evidence, it will be more likely to prove an attack relation, while the claim and evidence would show support relation. Negation words help to detect attacks relations. When the topic of the claim and the evidence is far from the warrant, diversion is necessary.

Table 5.5: Performance of fact checking in our proposed model and the impact of collecting the best warrant from multiple correct warrants, in addition to alternative warrant (rebuttal) or other irrelevant information.

Our Model	**Accuracy**
Fact predictor without warrant (given only claim-evidence pair)	73.95
Random **correct** warrant aware fact predictor	79.21
Best **correct** warrant aware fact predictor	81.69

Our findings support the hypothesis H2 that incorporating warrants can enhance the overall performance of a fact checking model.

5.5. Summary

Fact checking is the process of determining the truth of a claim. Detecting the truthfulness of a claim, as in fake news, using only existing knowledge of the news, e.g., evidence text, is generally insufficient, as the claim's rationale is implicit and is expected to be inferred by the reader, while it is necessary to comprehend the claim completely. The majority of previous models on this task used the claim and evidence as input, and the failure of the systems to detect the relationship resulted in poor performance, particularly for ambiguous data where some components, e.g., implicit premise, are missing.

To address the issue of poor performance, a model that can detect relationships based on previously extracted warrants from structured data is developed. For warrant selection, knowledge-based and style-based prediction models are combined to capture additional information that can be used to infer which warrant best bridges the gap between claim and evidence. Selecting a reasonable warrant may assist in resolving the evidence ambiguity issue if the proper relationship cannot be established. The experimental results indicate that incorporating the best warrant into the fact checking model improves fact checking performance.

Despite the critical nature of the implicit warrant required to comprehend the claim, there is a dearth of annotated argument justifications. The maximum dataset available for the ACRT, for example, is 1.7k samples, but deep learning models require a considerable amount of labelled data to train. Additionally, to maximise the benefits of deep learning algorithms for false information detection, the scarcity of labelled data for warrants must be addressed. Chapter 6 develops novel models to address the dearth of labelled warrants data.

Chapter Six: **Warrant Generation Models**

6.1. Introduction

Even though utilising a warrant can aid in the performance of fact checking tasks [383], to our knowledge, no previous work has proposed that a claim be connected to a piece of evidence via automated warrant creation rather than manual annotation. Additionally, no experiment was conducted using a labelled dataset, but rather through the use of case studies [159]. Unlike previous approaches that relied on structured annotated warrants [355] or manually generated warrants for emerging claims based on certain linguistic rules [159] that require a higher level of language comprehension and complex reasoning, our work is based on the automated generation of warrants for claims.

In chapter 5, we examined the use of warrants in fact checking, and our findings indicated that warrant consideration has a significant effect on the fact checking model's performance. However, the issue remains that the majority of current datasets lack sufficient annotations, posing a challenge for deep learning algorithms that require a large amount of labelled data to train. On the other hand, manually annotating massive amounts of noisy social media data for the purpose of fact checking is a time- and labor-intensive process.

This chapter will evaluate a variety of models to answer the research question RQ 3: how to generate high quality and more diversity warrants? For example, what contextual information can be incorporated into deep learning through the use of natural language processing techniques such as RST and causality for warrant generation? Is it possible to improve the diversity and quality of warrants by fine-tuning a pre-trained Language Model (BART) using Multi-Agent Network reinforcement learning? To our knowledge, this is the first time that the integration of reinforcement learning, and a generative adversarial network has been used to solve the warrant generation problem.

The remainder of this chapter is organised as follows. In section 6.2, we discuss our warrant generation models. Section 6.3 discusses the experiments and the findings, while section 6.4 summaries the work.

6.2. Warrant Generation Models

The overall framework of our warrant generation models, first and second, is shown in figure 6.1. This framework addresses the challenge of a lack of large, annotated data for warrant given claims and their evidence, the largest of which is ARCC [365] with less than 2k rows.

We develop three generator models; the first model trains a reinforcement learning agent to act as a generator, while the second model employs a reinforcement learning agent to enhance different generators via multi-head attention. The purpose of implementing these models is to determine which strategy produces the most promising results: using the RL agent as a generator or as a generator enhancer. The first model has two stages: the initial stage selects warrant-relevant fragments using various methods such as RST and causality, and the second stage selects warrant-relevant words to generate warrants via reinforcement learning agents. While the second model relies on RST and a deep learning mechanism to select candidate warrant relevant fragments, this model utilises a Multi-Head Attention Mechanism enhanced by reinforcement learning to generate warrants. Regarding the third model, the general architecture of our third proposed model for warrant generation begins at the bottom with the Bart model's implementation and ends at the top with the Multi-Agent Model's implementation, as shown in figure 6.4. We chose the BART model because of its encoder-decoder design, which makes it especially well-suited to constrained text generation, and because RL expands the range of applicants and methodologies available.

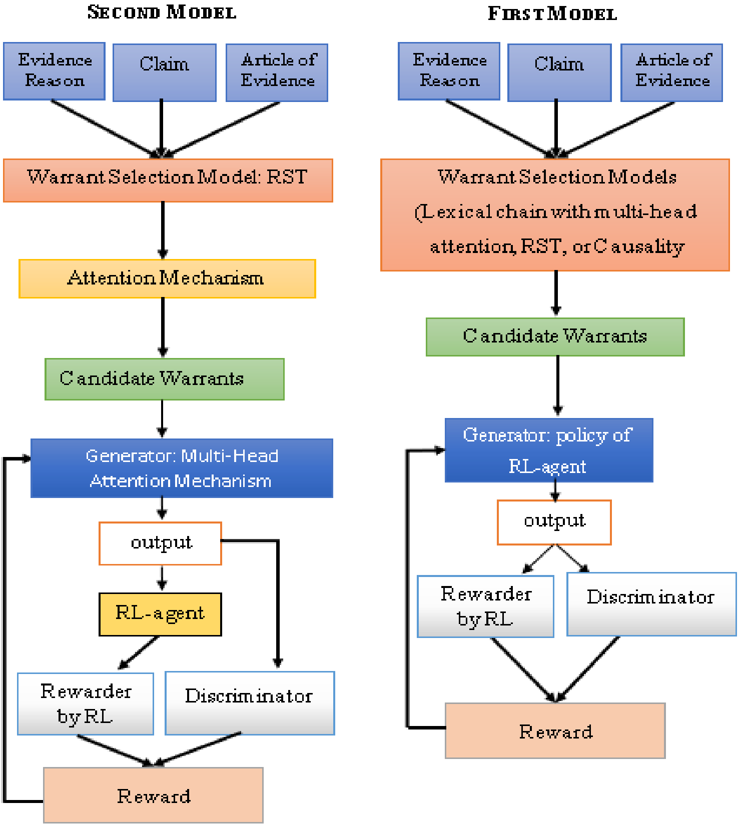


Figure 6.1: Warrant generation models

6.2.1. Model 1: Warrant Generation Using RL Generator

Models for warrant selection that identify pertinent sections of an unstructured text that contain the information required for a warrant.

6.2.1.1. The Initial Stage: Models for Identifying Warrant-relevant Fragments

The first stage in our warrant generation process is to select (retrieve) information that is pertinent to a claim and unstructured evidence. Increasing the efficiency of false information detection requires developing the ability to recognize the connection between a claim and a piece of evidence. Our proposed models include a Lexical Chain with Multi-Head Attention, an RST-based algorithm, and a Causality-based selection method, all of which are aimed at capturing more compelling reasoning warrants. Table 6.1 illustrates an example of the most pertinent information contained in a warrant in light of a claim and evidence which are highlighted in bold.

Table 6.1: An illustration of the task of locating the most pertinent information to the claim and supporting evidence from the ARCC [365].

Claim: "Greece will destroy the Euro Zone"
Evidence Reason (The premise is the specific piece of evidence included in the lengthy evidence): "Greece cannot support its own economy and is bringing the Euro down"
Article of Evidence: "The eurozone is now furiously bracing itself for the likely collapse of the Greek government. Faced with the prospect of Greece voting for a fully-fledged default and euro exit rather than last week's debt deal, the remaining eurozone members must themselves choose: stick even more closely together or be pulled apart. They will stick together – and survive. **However, the euro zone's survival has very little to do with Greece**. The Greek economy is too small to cause any noticeable impact on the eurozone and even the widespread and substantial financial contagion of a default can be absorbed. Last week's debt deal may not appeal to Greece, but the beefed-up bailout fund is capable of taking care of the immediate consequences of a Greek default. Containment has been addressed and would focus on supporting other indebted states. ***The euro zone's survival has little to do with Greece except to persuade other members to redouble their efforts and stick with the euro. The key reason for Greece continuing to play an important role in deliberations over the euro zone's future is that it highlights the question mark over member states' abilities to resolve the deep-rooted problems of poorly performing economies. The influence that Greece can still wield is a demonstration effect: If Greece leaves, will the result be disastrous or could the economy be galvanized into better performance, as those who favor exit appear to believe?*** "

6.2.1.2. Lexical Chain with Multi-Head Attention

Inspire by the data retrieval, question answering, and response selection models, a claim is viewed as a query and evidence as an appropriate document from which the candidate's responses should be selected. The lengthy text (as evidenced by ARCC) data will be condensed for warrant selection using the lexical chain model to retain the most informative words that are also the quietest to draw attention to the claim outputs (or a query).

We begin by detecting salient portions of text using Word Sense Disambiguation (WSD) and then extracting the lexical chains described in Al-Khawaldeh & Samawi [154]. In contrast to Al-Khawaldeh & Samawi [154], the proposed model attempts to select sequences from each cluster associated with the claim instead of selecting the sequences that are significant to different topics as in Al-Khawaldeh & Samawi [154]. For example, as in table 6.1, suppose we have "Greece will destroy the Eurozone," as the claim. To obtain the correct sense of the term ("zone"), its senses must be extracted at three levels. The first level extracts all possible senses for the "zone," the second level extracts the senses for these senses, and so on for the third level. The sense of a word refers to its meaning in a particular context.

The developed WSD algorithm consists of five steps as in Al-Khawaldeh & Samawi [154]:

1) Extract all the possible interpretations (senses) of each word in a sentence of evidence. Extract the three levels of senses for each sense. The first level is the senses of a word; the second level is the senses for each sense in the first level and so on.
2) Each word's senses are compared to the senses of all other words in the text and then establish connections between the related senses, a connection is established when there is a semantic relationship between the current word's senses and any other word's senses.
3) Calculate the strength of the connections.
4) Sum all the strengths of the connections.
5) Select the highest summation sense.

By empirically, the semantic relations and their associated weights are as follows:

- Repetition relation (same occurrences of the word), weight=1.
- Synonym relation (weight=1). In the example above, the word "zone" has a synonym semantic relation with the sense ("area")
- Hypernym and Hyponym relation (weight=0.5): Y is a hypernym of X if X is a (kind of) Y; X is a hyponym of Y if X is a (kind of) Y e.g., X="zone", Y=" ground".

- Holonym and Meronym relation (weight=0.5): holonymy relation is (the whole of), and meronymy relation is (part of). Y is a holonym of X if Y is a whole of X; X is a meronym of Y if X is a part of Y. X= "state", Y="zone".
- Gloss relation (definition and/or example sentences for a synset), (weight=0.5): consider the word="zone", gloss=" area having a particular characteristic".

Each sense has several weighted connections to other words' related senses. The weighted connections between the senses are added together. Lexical cohesion is used to differentiate between significant and unimportant sentences in a text. The text is segmented by lexical cohesion. Each segment consists of a series of sentences devoted to a single subject. Each word is assigned the correct sense after the proposed WSD algorithm is applied to the text above. Lexical chains (LCi) are formed by connecting the words' senses (meanings). If these senses have semantic relationships, then the words are related.

LC1:{money, account, transfer, cash, withdraw, bank}

LC2:{area, ground, region, segment, sector}

To begin, we use a Bi-RNNc to model the embeddings of claim words cl and chain words c, where $h^{c}_{i,1}$ denotes the hidden state of the t-th word in the i-th chain and $h^{cl}_{i,1}$ denotes the hidden state of the t-th word in the i-th claim. Following that, we perform an average-pooling operation on these hidden states, equation 6.1, to generate a vector representation of the i-th chain, equation 6.2.

Equation 6.1 $a_{vi} = avg\left(\left\{h^{c}_{i,1}, h^{c}_{i,2}, \ldots, h^{c}_{i,T^{c}_{i}}\right\}\right)$

Equation 6.2 $m_i = tanh\left(W_{cl} \cdot \left[a_{vi}; h^{cl}_{j,T^{cl}_{j}}\right] + b_{cl}\right)$

Mi can be thought of as a salience score for the i-th chain in the context of the claim representation, $h^{cl}_{j,T^{cl}_{j}}$.. The highest sigmoid output indicates the chain's importance concerning the claim; thus, the selected segment of evidence should be chosen based on this critical chain, which allows for the omission of irrelevant text. To model the relevancy of the segment of text towards the strongest chain, we first calculate the word alignment of the segment towards the chain. We use the embeddings of words in chain and segment to calculate the semantic alignment score as shown in equations 6.3 and 6.4:

Equation 6.3 $\mathrm{score}_{\mathrm{i,j,n}} = \mathrm{e}(\mathrm{A}^{\mathrm{c}}_{\mathrm{i}})^{\mathrm{T}}\, \mathrm{e}\left(\mathrm{A}^{\mathrm{s}}_{\mathrm{j,n}}\right)$

Equation 6.4 $maximum_{i,j} = max\left(\left\{score_{i,j,1}, \ldots, score_{i,j,T^{c}_{j}}\right\}\right)$

Where $e(A^{s}_{i})^{T}$ is word embedding in the segment, and $e\left(A^{c}_{j,n}\right)$ is word embedding in the chain, $score_{i,j,n}$ is the attention wei for the i-th chain word with the j-th segment word, s is a segment, c is the chain, n is the segment number, i is the index word of the segment, and j is the index word in the chain. The alignment score, maximum-i,j, is the weight assigned to the jth chain concerning the ith segment word. We take the highest attention weights from all scores and represent them as candidates' parts, retaining only the relevant parts. After selecting the most informative text from the evidence and obtaining reduced text, we will use multi-head attention to construct deep contextual representations for tokens located in different representation subspaces at different positions while preserving their syntactic form.

This model's general framework is divided into four steps.

- Apply word embedding for each word in the text.
- Use a BiLSTM and CNN to obtain the vector representation of the text.
- A multi-head attention mechanism that can capture relevant information from different subspaces
- Use the SoftMax layer for text classification to select the candidates' warrants.

The Elmo word embedding model will represent each word in each sentence as a deep contextual deep word representation. Elmo is a sophisticated, contextualised word representation that extracts the word's complex syntactic and semantic features [175]. On a variety of natural language processing tasks, including query answering and textual entailment, Elmo outperforms previous word embeddings such as word2vec and GloVe [147]. By reading each sentence in two directions: from beginning to end (forward) and from end to start (revers), we extract the most critical information and obtain contextual information about the current word using a CNN and a Bi-LSTM. The final encoded representation combines the Bidirectional hidden state representation and the Bidirectional hidden state representation.

Multi-head attention layer for claim-evidence text: A specific section of the text is critical in identifying the candidate warrant in a given claim-evidence. Numerous heads of attention assign each word the appropriate weight to represent the text's general semantics based on various factors. This work makes use of self-awareness to capture the relationship between the claim-evidence pair and the warrant.

In contrast to multi-head -attention from the literature, which typically considers V=K=Q and is derived from the same source, we define Q as each word in a candidate warrant is required to perform an attention calculation using all other claim words as keywords, where the warrant is a candidate sentence from the article. The attention layer receives three input texts: a claim text as a key, a candidate warrants as a query, and an evidence text as a value. Each of them contains a word vector containing all of the words in the input text. Multi-head refers to paying attention not only to the individual words in the sentence but also to the individual segments of the words. The vectors of words are divided into a fixed number of chunks (h, number of heads), and then multi-head attention is applied to the corresponding chunks, resulting in an h context vector for each word. The final values vector is created by concatenating all of that h to generate an encoded representation for each word in the input sequence (representation vectors) and add the word's attention score. The primary steps using the following example from the ARCC [365]:

- Candidate warrants from an article (query Q): “money will not be saved all the way around”
- Claim (keys K): “Privatization is a bad deal for cities and states.”
- Evidence (values V): “The only interest of the private sector is the bottom-line profits.”
  - The query is the input word vector for the Candidates warrants token, e.g., “money”.
  - The keys are the input word vectors for all of the claim's tokens [Privatization, is, a bad, deal, for, cities, and, states]
  - The query's word vector is then DotProducted with the word vectors of each key, yielding n numbers, i.e., "weights." Following that, the weights are scaled.
  - The weights are then subjected to a ‘SoftMax' operation, which normalises all weights to values between 0 and 1.
  - Finally, the input word vectors, e.g., values, are summed in a "weighted average of the value vectors using the previously normalised weights. It generates a single output word vector representation of the Candidates warrants word, as in equations 6.5 and 6.6:

Equation 6.5 $$\mathrm{Attention}\ (\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V}$$

Equation 6.6 $$\mathrm{head_i} = \mathrm{Attention}\left(\mathrm{QW_i^Q}, \mathrm{KW_i^K}, \mathrm{VW_i^V}\right)$$

  - All word vectors are getting similarly; the attention mechanism is applied to all word vectors. Single output word vector representation of “Privatization” is finally obtained and so on for all words, resulting in o, output word vector representation, as shown in equation 6.7:

Equation 6.7 $$\mathrm{O} = \mathrm{MultiHead}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{Concat}(\mathrm{head_1}, \dots, \mathrm{head_h})\mathrm{W^o}$$

O is the output of multiple attention functions used in multi-head attention capturing explicit and implicit patterns. It converts Q, K, and V subspaces to C subspaces using various learnable linear projections. To capture various contexts, information from various representation subspaces at various positions can be prioritised. Each head generates an attention distribution to its subspace to represent the final state when all attention heads are considered. The independent operation's result is then spliced into a linear transformation. To obtain the multi-head attention result M, as in [212]. We construct an auxiliary feature vector from the topic T and sentiment vectors S; the concatenated features are TS. Assuming that those features are consistent across inputs, we combine them with the output of multiheaded attention O to create a new representation, Onew=O+TS; all words vectors are concatenated as S= Onew1, Onew2…. Onew n. Then, using a SoftMax layer as an activation function, classification is performed. Thus, the probability of current candidates warrants Y, as shown in equation 6.8:

Equation 6.8 $\mathrm{Y} = \mathrm{softmax}(\mathrm{W} * \mathrm{S} + \mathrm{b})$

6.2.1.3. RST-Based Algorithm

Due to the causal and semantic relationship between claim, evidence, and warrants, we were inspired by RST's discourse analysis, which identifies a rhetorical relationship between two text spans, nucleus and satellite, where the nucleus contains more informative text than the satellite, which contains additional information. Given that warrant provides reasoning for a claim in the form of cause, purpose, motivation, and circumstance, in our model, the nucleus (span) of the RST relation is matched against the claim and the relationship (primarily implicit or explicit causal) with the satellite; the best candidate warrant is determined by the most pertinent RST relation between the claim and the warrant span discourse units.

RST can be used to describe the relationships between text's internal components. RST relations divide the text into rhetorically related segments that may be further divided, resulting in a hierarchical rhetorical structure. Each segment corresponds to a nucleus or satellite. It demonstrates that coherence relations can have a beneficial effect on both the claim and the justification. For instance, the nucleus contains an idea that the author regards as the nucleus.

We will use RST to conduct discourse analysis, which identifies rhetorical relationships between two text spans: nucleus and satellite, with the nucleus containing more informative text than the satellite, which contains additional information. Numerous RST relationships may aid in the exploration of the information included in text, as illustrated in table 6.2. We give the data in tabular style to facilitate organisation. Because a warrant justifies a claim, it serves as the cause, purpose, motivation, and circumstance. The nucleus (span) of the RST relation is matched against the claim and the relationship (primarily implicit or explicit causal) with the satellite in our model; the best candidate warrant is determined by the most relevant RST relation between the claim and the warrant span -discourse units. Heilman & Sagae's work will be used to implement RST [384]. An example of a nucleus or satellite, where the claim "I believe the weather is cold and wet" is the nucleus and the supplementary text "since the temperature has decreased by 15 degrees Celsius" is a satellite, connected with the *explanation* rhetorical relation. In this example, the satellite clause explains the nucleus, as in argumentation model such as Toulmin model, the warrant is supplementary for main information, claim, so our work considers warrant is satellite and claim are the nuclei.

Table 6.2: Organization of the relation definitions [150]

- Circumstance
- Summary
- Elaboration
- Background
- Enablement and Motivation
- Relations of Cause

- Antithesis and Concession
- Antithesis
- Concession
- Condition and otherwise
- Condition
- Restatement

- Enablement
- Motivation
- Evidence and justify
- Evidence
- Justify
- Purpose

- Otherwise
- Interpretation and Evaluation
- Interpretation
- Evaluation
- Restatement and Summary
- Sequence

Based on this complementary relationship between satellites and nuclei, we argue that certain words in certain nucleus-satellite relationships may be more significant than others, e.g., they indicate the clause has a warrant. Thus, we argue that a satellite should be considered when determining a warrant in a case where the satellite is linked to the claim's nucleus. On the other hand, we argue that the nucleus does not contribute to the satellite's understanding. Thus, words contained within a satellite differ from those contained within a nucleus, as in figure 6.2:

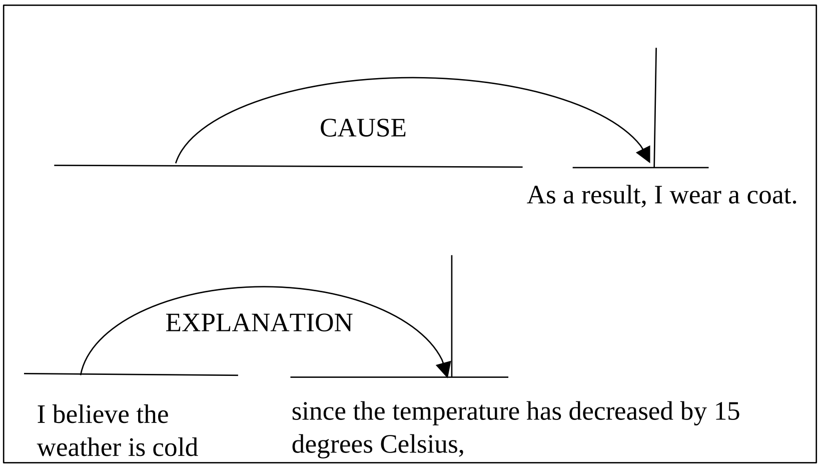

CAUSE
As a result, I wear a coat.
EXPLANATION
I believe the weather is cold
since the temperature has decreased by 15 degrees Celsius,

Figure 6.2: The relation between a nucleus and a satellite, an example of nucleus or satellite, with RST relation

The RST-based algorithm to select a warrant for a claim is as follows:

1. Input: evidence text, claim query, query expansion
2. Result: warrant
3. Begin
4. Segment texts to clauses based on cure phrases (connectors words)
5. Find rhetorical relations between the clauses to build all RS-trees for evidence text
6. Check the rhetorical relations between the segments: nucleus and satellite, e.g., explanation, interpretation, result or justification.
7. If a segment is a nucleus and is relevant to claim query or query expansion, then the satellite is a warrant and vice versa.
8. Save as candidate part of the warrant and continue to the next candidate warrant
9. End

6.2.1.4. Causality-Based Selection

The causal relationships provide knowledge that allows for the interpretation of the evidence-based claim. As the warrant explains how the data leads to the claim, it is necessary to recognise causalities expressed explicitly in answer phrases such as "because" and to use those recognised causalities as a guide for locating proper answers. Causalities expressed in one text may be expressed with explicit cues in other texts. in the form of texts expressing causal relationships (e.g., "[Tsunami occurred] effect as a result of [a sudden displacement of sea water] cause"). If we can identify causal relations in which the effect part corresponds to a target why-question, the cause parts may contain useful information for generating appropriate compact answers, such as important keywords to include in the compact answers. We retrieve causal relation expressions that are relevant to claim C, such as effect and cause relevant statements, given a target claim C.. Thus, we automatically extract causal relations relevant to a target why-question from the web, such as "[Microsoft's machine translation has made significant progress in recent years] effect since [it began using deep learning] cause":

Because the warrant has a casualty and a reason, we used a why–how to approach in our work. A contrast relationship implies adversarial justification (rebuttal). The event causes demonstrate what occurs (effect) in a claim and a warrant. Table 6.3 illustrates several of these relationships and the position of claim and warrant and evidence. The presence of causality is checked in a sentence, where causality refers to the relationship between cause and effect in a sequence of events. Oh, et al. [385] suggested causality-attention: A convolutional neural network with multiple columns for why-QA.

Table 6.3: Examples of causality relations

- Claim ***as a result of*** warrant and evidence
- ***because of the*** warrant and evidence, the claim
- warrant and evidence ***Consequently*** claim
- ***due to*** warrant and evidence, the claim
- ***due to the fact*** warrant and evidence, the claim
- ***on account of*** warrant and evidence, the claim
- ***seeing that*** warrant and evidence, the claim
- *Claim* **So** warrant and evidence
- the claim ***as a consequence of*** warrant and evidence,
- warrant and evidence ***the reason, claim***
- warrant and evidence ***therefore claim***
- ***for this reason,*** warrant and evidence, the claim
- warrant and evidence ***this led to claim***
- ***this cause*** warrant and evidence, claim
- ***in order to*** warrant and evidence, the claim
- warrant and evidence, the warrant and evidence ***resulting in*** the claim
- warrant and evidence ***Thereby claim***
- warrant and evidence ***Similarly claim***

The claim expansion process in our work is inspired by (question query Q) [386]–[388], which employs a word embedding to expand the query (in our work, claim) and wordnet expansion [389]. The model checks for hypernyms, such as food, and hyponyms, such as fruit, in addition to meronyms and holonyms; a branch is a meronym (part meronym) of a tree, whereas heartwood is a meronym (substance meronym) of a tree, and the forest is a holonym (member holonym) of a tree. If the evidence text has causality with the claim or is highly semantically related to the claim (more connected to the claim), those texts will receive additional scores as part of the candidate's warrant.

Along with the most closely related parts by wordnet relation, two types of attention mechanisms will be used to score the candidates' warrants: similarity-attention [390] and causality-attention [385]. The similarity-attention mechanism calculates the cosine similarity between the embeddings of claim and evidence text to generate an attention feature vector for evidence words. In contrast, causality attention focuses on evidence words causally related to claim words and is used to generate causal embeddings focusing on causal relations to generate a causality attention feature vector. When confronted with passages containing possible causes/reasons for a given claim, causality attention can be focused on words and their contexts. The matrix of causality-attention features is constructed using scores indicating the degree to which two words are causally related (one in a claim and another in a warrant passage).

6.2.1.5. The Second Stage: RL for Identifying Warrant-Relevant Words

Candidate warrant selection techniques will be analysed to ascertain the warrant's scope (to retrieve the warrant). We propose to collect significant, warrant-relevant words from a lengthy fragment using reinforcement learning RL (through actions). RL shows a promising result in different methods [391]–[393] where the model acquires knowledge through interaction with its environment and is rewarded for completing tasks. In [394], text generation is formulated as the sequential decision-making problem

Due to the discrete nature of the data and no gradient can be obtained, we use RL to guide our sequential decision policy network's training and use lexical in nature measures for evaluation a reward function, for example, rouge or BLEU. We hypothesise that a sequential decision policy network can aid in the detection of warrants. A delayed reward is used to direct the policy's learning process based on the interaction of predicted and actual warrants. As illustrated below, we briefly discuss state, action and policy, motivation, and objective function.

Given a candidate warrant's word sequence $\mathrm{w_i}, 1, \dots, \mathrm{w_i}, \mathrm{k_i}$ the policy network *πl* attempts to select the warrant-relevant word $\mathrm{w_i}, \mathrm{j}$ and eliminate irrelevant ones. The policy network employs a stochastic policy to check the probability of an action at each state, and it learns through delayed reinforcement after the sequence of actions is completed. We construct the policy *πl* for selecting words over a word sequence using the Bi-GRU model. We use Bi-GRU because it has fewer parameters than LSTM and thus performs more quickly with efficiency [395].

State (st): given the claim, evidence and candidate warrant as input, the policy aimed to decide the warrant relevant words as delete, keep or generate. Afterword embeddings $e_i$is performed, we use Bi-GRU to get the vector representation of candidate warrant $h_s^{(1)} + h_s^{(1)} + h_s^{(2)} + \cdots + h_s^{(n)}$. Following the acquisition of claim and evidence hidden state representations, we then pool the vectors on an average basis $claim^{(l)}$ and $evidence^{(l)}$ through equations 6.9 - 6.13:

Equation 6.9 $\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)} = bGRU\left(e_i, \vec{h}_{i-1}^{(1)}, \overleftarrow{h}_{i+1}^{(1)}\right)$

Equation 6.10 $h_i^{(1)} = W_1\left[\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)}\right]$

Equation 6.11 $claim^{(l)} = \frac{1}{N-1}\sum_j h_j$

Equation 6.12 $evidence^{(l)} = \frac{1}{m-1} \sum_j h_j$

Equation 6.13 $st = h_s^{(n)} + claim^{(l)}\ evidence^{(l)}$

To produce a vector representation for both, claim and evidence, we use average-pooling operation over hidden states as shown in equations 6.14 and 6.15.

Action: A stochastic policy uses state information for deciding to select the current word or not. We adopt a logistic function (conditional probability) to decide whether this word is relevant for a warrant or not, as in equation 6.14.

Equation 6.14 $action = sigmoid(W * st + b)$

Reward-1: We employ attention mechanisms at each stage of text representation, the actual warrant and predicted warrant. By assigning weights to encoding vectors, it is possible to highlight specific parts of the input that are more important for detecting warrants, candidate warrant CW, and actual warrant AW similarity, as in equation 6.15 – 6.20.

Equation 6.15 $u_{ij} = tanh\left(W_w \cdot \left[h_{ij}; CW\right] + b_w\right)$

Equation 6.16 $a_{ij} = softmax\left(u_{ij}\right) = \frac{exp(u_{ij})}{\sum_{t=1}^{N} exp(u_{it})}$

Equation 6.17 $CW = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij}$

Equation 6.18 $u_{ij} = tanh\left(W_w \cdot \left[h_{ij}; AW\right] + b_w\right)$

Equation 6.19 $a_{ij} = softmax\left(u_{ij}\right) = \frac{exp(u_{ij})}{\sum_{t=1}^{N} exp(u_{it})}$

Equation 6.20 $AW = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij}$

Finally, reward guides the policy regarding the selection of warrant-relevant words within a warrant sequence. We use the connection of vectors and the SoftMax function to combine the predicted warrant CW a representation and the actual warrant AW representation for similarity classification, as in equation 6.21:

Equation 6.21 $Y = SoftMax(W[CW \oplus AW] + b)$

Semantic coherence Reward 2: the generated warrant to check if it is grammatical and coherent as in equation 6.22:

Equation 6.22 $r_{SC} = \frac{1}{N_y} log\, P_{seq2seq}\ (y|x_i) + \frac{1}{N_{x_1}} log\, P_{backward-seq2seq}\ (x_i|y)$

Pseq2seq denotes the likelihood of the seq2seq model (the probability of generating the predicted warrant given the previous warrant). Pbackward seq2seq denotes the backward probability of actual warrant given the current generated warrant.

In previous work [396], we trained separate models (single agents) to locate the warrant given a claim and evidence. The first model employs Lexical Chains, as proposed by Al-Khawaldeh and Samawi [154], which aid in extracting the most informative words and thus reducing the text's size. After obtaining the summarised text, the claim's related fragments and evidence are captured using the multi-head attention model. The second model employs a Rhetorical Structure Theory-based algorithm to segment each text into two spans, nucleus and satellite, with a higher probability of being nucleus. Finally, the causality model: because the warrant possesses a causal and rational nature, the causality relations denote the text fragments that contain one of the following relations: justification, interpretation, or confirmation. These are more extraction-oriented models than generation-oriented models. As a result, our model attempts to generate warrants by combining multi-head attention theory and rhetorical structure theory.

6.2.2. Model 2: Warrant Generation Using a Multi-Head Attention Mechanism Generator Enhanced by RL

In model 1, we use a reinforcement learning agent as the generator, whereas in model 2, we use reinforcement learning as an enhancer for the generator to determine which is more effective. We develop justifications for why an argument is persuasive, discovering that adding word embedding features improves performance. Given a claim c = c_1; c_2; …; c_k containing k words, and an evidence d = d_1; d_2; … d_n consisting of n words, the objective is to generate a warrant for the context y = y_1; y_2; … $_{ym}$ containing m words. The objective is to find an output Y* that maximizes the probability p(Y| c ; d), Y is the warrant, and c and d are claim and evidence, respectively.

The RST based algorithm is used to locate a warrant for the claim. We take each word as input to get the claim embedding vectors as in equation 6.23.

Equation 6.23 $e_c = \{e_c^1, e_c^2, e_c^3 \dots e_c^n\}$

Similarly, the candidate warrant is also embedded as vectors as in equation 6.24.

Equation 6.24 $e_w = \{e_w^1, e_w^2, e_w^3 \dots e_w^m\}$

Then we apply cosine similarity to compute the final score as the relevance of a claim to a warrant to detect the candidates' warrants: score (claim, candidates warrant) = cosine similarity (e_c, e_w). The highest score means that it is more likely that the warrant is plausible. The model adopts BiGRU to represent both claim rc and candidate warrant rw because it operates well in learning long term dependencies and is fast in training.

To reduce the spatial size of the representation and retain essential features, we adopt mean pooling to calculate the claim $mcl^{(cl)}$, evidence $mev^{(ev)}$ and warrant $m^{(w)}$ pooling vectors through the equations 6.25 – 6.27:

Equation 6.25 $mcl^{(cl)} = \frac{1}{N-1}\sum_i \ h_{claim}^{(i)}$

Equation 6.26 $mev^{(ev)} = \frac{1}{M-1}\sum_i \ h_{evidence}^{(i)}$

Equation 6.27 $m^{(w)} = \frac{1}{K-1}\sum_i \ h_{warrant}^{(i)}$

We define the attentive representation of claim, evidence, and warrant to one another, i.e., the attentive representation of the effect phrase concerning the cause phrase, to consider the score and impact of each of them on the other, as follows:

The claim representation with its candidates' warrants clw_t as the equations 6.28 – 6.30:

Equation 6.28 $a_{t,i}^{cl} = v_{cl} \cdot \ tanh\left(W_1 m^{(warrant)} + U_{cl} h_i^{claim}\right)$

Equation 6.29 $\alpha_{t,i}^{cl} = \frac{exp(a_{t,i}^{cl})}{\sum_{i=1}^{|cl|} \ exp(a_{t,i}^{cl})}$

Equation 6.30 $clw_t = \sum_{i=1}^{|cl|} \ \alpha_{t,i}^{cl} h_i^{claim}$

The candidates warrant representation with their claim wcl_t as in equations 6.31 – 6.33:

Equation 6.31 $a_{t,i}^{w} = v_{w} \cdot \ tanh\left(W_2 m^{(claim)} + U_{w} h_i^{warrant}\right)$

Equation 6.32 $\alpha_{t,i}^{w} = \frac{exp(a_{t,i}^{w})}{\sum_{i=1}^{|w|} \ exp(a_{t,i}^{w})}$

Equation 6.33 $wcl_t = \sum_{i=1}^{|w|} \ \alpha_{t,i}^{w} h_i^{warrant}$

The evidence representation with its candidates' warrants evw_a^t as in equations 6.34 – 6.36:

Equation 6.34 $a_{t,j}^{ev} = v_{ev} \cdot tanh\left(W_{ev} m^{(warrant)} + U_d h_j^{evidence}\right)$

Equation 6.35 $a_{t,j}^{ev} = \frac{exp\left(a_{t,j}^{ev}\right)}{\sum_{j=1}^{|ev|} exp\left(a_{t,j}^{ev}\right)}$

Equation 6.36 $evw_a^t = \sum_i \ a_{t,j}^{ev} h_{a,i}^{(evidence)}$

The candidates warrant representation with its evidence wev_a^t as in equations 6.37 – 6.39:

Equation 6.37 $$a_{t,j}^{w} = v_w \cdot tanh\left(W_w m^{(evidence)} + U_d h_j^{warrant}\right)$$

Equation 6.38 $$a_{t,j}^{w} = \frac{exp\left(a_{t,j}^{w}\right)}{\sum_{j=1}^{|w|} exp\left(a_{t,j}^{w}\right)}$$

Equation 6.39 $$wev_a^t = \sum_i \; a_{t,j}^{ev} h_{a,i}^{(warrant)}$$

Finally, we combine all these representations for *causal/noncausal in* equation 6.40:

Equation 6.40 $$Y = softmaxY(clw_t + wcl_t + evw_a^t + wev_a^t)$$

Causal/noncausal Y means the candidates warrant either plausible or not.

Multi-Head Attention Mechanism with Multiple Heads: This model employs the transformer network [212], which is based primarily on deep learning and dot products and is composed of fully connected layers from both the encoder and decoder. It replaced recurrence or convolution with the multi-head -attention transformer's encoder, composed of six identical layers, each of which is composed of two sub-layers: a multi-head -attention mechanism and a position-wise fully connected feed-forward network [397]. A residual connection and layer normalisation are used to generate outputs from two sublayers. The transformer Decoder is also composed of a stack of identical layers to the encoder, except that it includes a third sublayer that implements a multi-head attention mechanism over the encoder's output, as illustrated in figure 6.3.

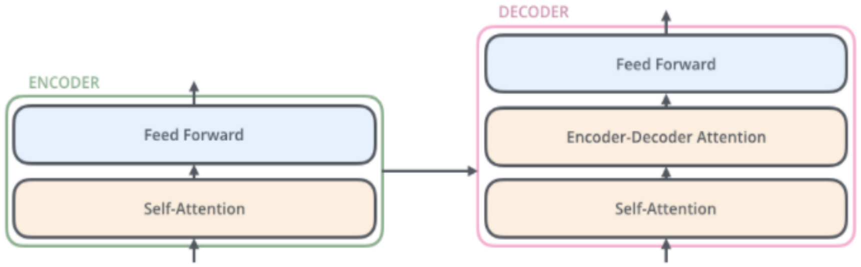


Figure 6.3: Transformer encoder-decoder architecture [397]

To capture the relationship between words in various positions, it computes the relevance of a set of values (information) using the same attention mechanism. In practice, the attention function is computed concurrently on a set of queries. It computes the attention function for a matrix Query, Keys, and Values that contains a collection of queries, keys, and values. Each head corresponds to a layer of attention [397]. The encoder converts a sequence of discrete representations in the form X = (x1;... xh) to a sequence of continuous representations in the form z = (z1; ... zh). In our work, X refers to the claim, evidence, and the average embedding of selected warrants used to generate warrants. the decoder then generates an output sequence consisting of one element at a time (y1;... yh). For the multi-head attention mechanism, h = 8, implying the use of eight parallel attention layers. To ensure the model's sequence, positional encoding is added to the input embeddings at the end of the encoder and decoder stacks. It can use embedded vectors to represent the relative positions of each sentence's words and then combine them with the sentence embeddings, as in equations 6.41 and 6.42:

Equation 6.41 $Z_i = head_i = \ attention\left(QW_i^Q \ , KW_i^K \ , VW_i^V\right)$

Equation 6.42 $Z = MultiHead(Q, K, V) = Concat(Z_1, \ldots, Z_h)W^o$

Our model takes as input a claim concatenated with candidate relevant warrants. After applying word embeddings, W-emb., to input words, we use The BiGRU to capture semantic information about past and future words. BiGRU utilises a forward and backward LSTM as encoder hidden layers to determine the hidden state of the time step t ht. Then, as in Vaswani et al. [212], we use residual connection around the output of the Bi-GRU layer to stabilise the model's training, followed by layer normalisation, as equation 6.43:

Equation 6.43 $h_t^* = LayerNorm(W_{emb} + h_t)$

Final encoder layer output H is the output of the add and Norm layer, equation 6.44.

Equation 6.44 $H = (h_1^*, h_2^*, \ldots, h_i^*, \ldots, h_n^*)$

We compute a representation of the sequence using multi-head attention, which is an attention mechanism associated with the various positions of a single sequence. The attention distribution is calculated as follows: Output H is Query vectors, keys vectors K2, and values vector Ve. The encoder's attention module is largely based on Vaswani et al.'s multi-head attention [212], as in equations 6.45 – 6.47:

Equation 6.45 $e^t = \frac{Q^e K^{eT}}{\sqrt{D}}$

Equation 6.46 $a^t = softmax(e^t)$

Equation 6.47 $Attention(Q, K, V) = a^t V^e$

The multi-head attention adjusts the Q, K and V matrix dimensions by h different linear layers to h queries, keys, and dimension values. The linear transformation parameters W of Q, K and V, are different each time based on the learnable parameter's matrix for the $head_s$. Then, h parallel heads are used to concentrate on distinct semantic spaces. The result of the independent operation is spliced into a linear transformation to obtain the result ce of multi-head attention, as in equations 6.48 and 6.49:

Equation 6.48 $head_i = attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$

Equation 6.49 $ce = MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^o$

Then decoder d generates word by word based on:

- The encoder e attention context ce is the output of multi-head soft-attention of sequence words input.
- The recurrent attention context, c_t^{ed}, it is based on each hidden state s_t of the decoder as query and hidden state output of the encoder as keys -values vectors of multi-head -attention.
- The decoder attention context c_t^d, Where multi-head-attention of all the predicted tokens is used.
- The decoder hidden state s_t. (equation 6.50) and the vocabulary probabilities (equation 6.51)

Equation 6.50 $s_t = GRU\left(s_{t-1}, Y_{t-1}, c_{t-1}^{ed}\right)$

Equation 6.51 $Pv = softmax\left(W'\left(W\left[c_t^e, c_t^{ed}, c_t^d, s_t\right] + b\right) + b'\right)$

c_t^{ed}t is the output of multi-head soft attention. The decoder has an embedding layer, a unidirectional GRU and a SoftMax layer. We use the hidden states of the decoder layer and the final encoder layer output H for obtaining the attention context c_t^{ed}. Besides feeding the attention context to all decoder GRU layers, we also feed it to SoftMax. This is important for both the quality of our model and the stability of the training process.

An encoder-decoder LSTM or GRU network is used to automatically approximate internal states and formulate potential actions for the reinforcement learning agents Sarsa or DDQN. The RL agents take the decoder output at time t as input and estimate each action's advantage values that learn to select an action (e.g., a word) from a list of possible actions to improve the current warrant sequence. For Sarsa, because it is learning an action-value function rather than a state-value function, it differs from Q-learning in that it does not require using the maximum reward for the next state. However, Deep-Q Networks is Q-learning with a deep neural network function that employs an epsilon-greedy policy to select actions for the Q-network approximator. Each decoding iteration will modify the current SARSA or DDQN by predicting which actions should be taken to accumulate a larger long-term reward.

6.2.3. Model 3: Extension for Warrants Generations Using a Language Model and a Multi-Agent System

Each argument begins with a claim, which is followed by one or more premises supporting the conclusion. The warrant is a critical component of Toulmin's argument model; it explains why the premises support the claim. Despite its critical role in establishing the claim's veracity, it is frequently omitted or left implicit, leaving readers to infer. We consider the problem of producing more diverse and high-quality warrants in response to a claim and evidence.

To begin, we employ BART [398], the most recent pre-trained language model for text generation, as a conditional sequence-to-sequence language model to guide the output generation process. On the ARCC dataset [365], we fine-tune the BART model. Second, we propose the Multi-Agent Network for Warrant Generation as a model for producing more diverse and high-quality warrants by combining Reinforcement Learning (RL) and Generative Adversarial Networks (GAN) with the mechanism of mutual awareness of agents. In terms of warrant generation, our model generates a greater variety of warrants than other baseline models. The experimental results validate the effectiveness of our proposed hybrid model for generating warrants.

This section describes the two-step process we use to generate a warrant for a specific claim and its premise. To begin, we generate warrants from the argument's claim and premises, utilising a pretrained language model that has been fine-tuned for this task. Then, the generated warrants are then fed into a multi-agent system to enhance the quality of modified versions of the input warrants. Figure 6.4 depicts the overall architecture of Our Proposed Model for Warrant Generation.

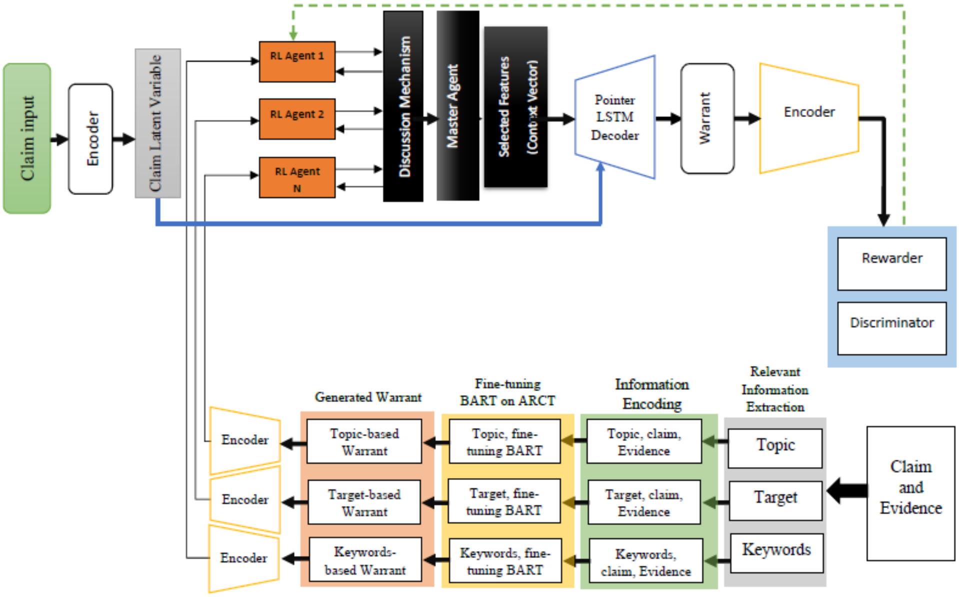


Figure 6.4: The general architecture of our third proposed model for warrant generation

6.2.3.1. Extraction of Relevant Information

This section discusses our relevant strategies to supplying the necessary kinds of information for generating an informative warrant, such as the topic, targets, and keywords.

- **Topic**: The topic of an argumentative text is a brief description of its subject. We use the associated debate title as the topic from the annotated ARCC data.
- **Target**: We employ Alshomary et al.'s approach [335], which introduced the concept of target extraction by concentrating on the inference of a conclusion's target.
- **Keywords**: It can be considered of as representations fed into the decoder along with the input sequence representation. Using a model of neural-based keyword extraction techniques, we identify the keywords for each argument in the corpus BiLSTM [399].

6.2.3.2. Fine-tuning BART on ARCC

To train our generation model, we use BART [398] , a pre-trained conditional language model that makes use of an auto-regressive transformer [212], [398]. The initial task is to elicit pertinent information from a claim and its premise. We then fine-tune BART using the ARCC data. To accomplish this task, we concatenate claim and premise as input to the BART encoder using the special delimiter "SEP" in addition to the extracted knowledge. To promote greater diversity and quality in our generated warrants, we generate three distinct versions of warrants that take into account a variety of relevant data as input sequence encoding of various pieces of knowledge: the topic, target, and keywords.

6.2.3.3. Multi-Agent for Warrant Generation Model

In general, a reinforcement learning (RL) network is a robust Markov Decision Process (MDP) model that maximises a numerical reward signal from a teacher in order to solve complex machine learning problems [225]–[227]. We employ reinforcement learning agents, such as the Deep-Q Network (DQN), in our model to help generate more informative (attention) features and to correct the decoded generated output for the generative model by enriching the GAN model with additional contextual information. Numerous reinforcement learning agents were used in our experiments to train the generator by feeding it with correct representations, including Double DQN [400], [401], State-Action-Reward-State-Action (SARSA) [225], Cross-Entropy Method (CEM) [402], [403].

Typically, our model encodes the BART-generated warrants using multiple local encoders, namely GRU, each of which is dedicated to a particular piece of knowledge. At first, each agent acts independently; agents learn to choose the appropriate action for the current time step, for example, which words or features to select. Following that, each agent updates their actions (selected features) in response to other agents' mutual awareness, e.g., by averaging the outputs of other agents' decisions, with each agent's final hidden state output being sent as additional input in the form of a message. The generator will produce better vectors as a result of receiving improved input from the multi agent's encoder. The reinforcement learning agent selects the most informative elements for the GAN generator, which generates a more informative feature vector. Following that, the feature vector is passed to the decoder in the pointer generator network, which generates an output that can fool the discriminator. Finally, the discriminator examines the decoder's final informative feature vector, which is capable of discriminating between real and generated token text. while the rewarder can provide rewards to the reinforcement learning agent during training.

Mutual Awareness of Agents: To assist the decoder in selecting the most appropriate sequence features, each agent should consider the information of other agents by sending a message to the shared representation room and receiving it to update each agent's context vector. A reinforcement learning architecture is when an agent attempts to maximise the reward associated with a particular action based on its observation. An agent chooses an action based on environmental observations and is rewarded. The objective is to find the optimal policy that chooses the action that maximises reward. The agent may act collaboratively or independently. The agents, multi-agent systems collaborate and associate in order to increase the model's overall utility. This paper introduces the multiple coder agent, in which each agent generates its representations of the data it receives. Then, through the shared representation room, all agents share information.

In our model, each agent encoder generates a representation for its corresponding input. We apply different encoder functions with different inputs, where each encoded representation represents the distribution of data. Generally, each agent will average the outputs of other encoder agents, which are conditioned on the information received from them (last hidden state output), as $v^{(k)}$. A multi-agent communication mechanism occurs in the shared representation room. After the agent's encoder function makes their encoding independently, in the next step, it passes them to the shared representation room based on a fine communicated mechanism. The shared representation room gathers other agents' outputs to update their own encoding decisions, which later enhances decoding decisions.

Each agent takes the encoded information $h_i^{(k)}$ from its encoder, which represents a particular view. It considers other agents' information by averaging the last hidden states of other encoders $h_{m,I}^{(k)}$, to produce other important information $v^{(k)}$. An attention vector

$f\left(h_i^{(k)}, v^{(k)}\right)$ is produced by considering its encoded feature $h_i^{(k)}$, previous decoder state s^{t-1} and other $v^{(k)}$. Finally, the context vector $c_{tj}^{k}$is updated based on attention distribution a_{tj}^{k}. The steps are as follows:

- Fetch messages from the shared representation room, as in equation 6.52:

Equation 6.52 $\mathrm{v}^{(\mathrm{k})} = \frac{1}{\mathrm{M}-1} \sum_{\mathrm{m}\neq\alpha} \mathrm{h}_{\mathrm{m,I}}^{(\mathrm{k})}$

- Update context vector for each agent, as in equations 6.53, 6.54 and 6.55:

Equation 6.53 $f\left(h_i^{(k)}, v^{(k)}\right) = v_1^T \, tanh\left(W_3 h_i^{(k)} + s^{t-1} + W_4 v^{(k)} + clt\right)$

Equation 6.54 $a_{tj}^{k} = \frac{exp(f_{tj})}{\sum_{k=1}^{l} exp(f_{tk})}$

Equation 6.55 $c_{tj}^{k} = \sum_{j=1}^{n} a_{tj_i}^{k} h_i$

W_{n} are parameters of weights

- Update shared representation room: shared representation room matrix initialized by zero vectors, then all attention context vectors of all agents are concatenated into this matrix. Transfer the updated context vector to the shared representation room.
- Finally, each time the decoder input has one of these context vectors.

With regard to the encoder for each auxiliary input aux (topic, target and keywords), we use Gated Recurrent Units (GRU), $h_i^{(c1)}$ for encoding the aux. It reads the aux and computes a hidden representation for each time step. Concerning the attention mechanism for the aux, the decoder generates an output word at each step by focusing on different aux portions. We begin by describing the claim attention model, which uses equation 6.56 and 6.57 to assign weights to each word in the aux at each decoder time step.

Equation 6.56 $a_{t,i}^{cl} = v_{cl} \cdot tanh\left(W_{cl} s_t + U_{cl} h_i^{cl}\right)$

Equation 6.57 $\alpha_{t,i}^{cl} = \frac{exp(a_{t,i}^{cl})}{\sum_{i=1}^{|cl|} exp(a_{t,i}^{cl})}$

St is the decoder's current state at time step t (we will see an exact formula for this. The final aux representation at time step t is computed as equation 6.58:

Equation 6.58 $cl_t = \sum_{i=1}^{|cl|} \alpha_{t,i}^{cl} h_i^{cl} \cdot$

The Master Agent: To use multiple new context vectors generated by agents and updated in real-time by the shared representation room. Because we have N context vectors, each of them considers it to contain local information in addition to global information gathered from other agents. We use max and mean to obtain the generator's final context vector via the master agent. Additionally, three matching methods are used to extract the generator's various inputs from context vectors c_{t-1}^{*}:

1. Concatenation of individual representations of all-new context vectors sent to the generator.
2. Element wise product of all new context vectors sent to the generator.
3. The absolute element-wise difference of all-new context vectors sent to the generator.

The outputs of all methods (maximum, mean, concatenation, element-wise product, and absolute element-wise) are connected via a fully connected neural network, which serves as the input distribution, and concatenated to controllable information C fed the generator. C is the general context vector generated by the master agent, a combination of an actor and a critic trained to select the optimal context vector.

Conditional Variational Autoencoder for Claim: For the claim, variational autoencoders are used to obtain the compressed feature vector representation, and the distilled information is used to train the generator to generate a new generated warrant more real toward the claim distribution.

The concept behind Variational Autoencoders (VAEs) is to generate texts using a generator based on encoded data (latent space), where the posterior and prior of the encoded data are tuned to minimise KL divergence loss. We aim to capture the fundamental and complex semantic structures underlying the warrants generated by our model. To accomplish this, we propose the use of conditional VAE, a variant of the VAE.

Our model extracts the representation's unstructured part z using BiGRU as an encoder for claim input. We consider incorporating topic information into latent variables as a guide for generating sentences that fall under the target's stance category (prior category vector is concatenated at each generation step by the decoder to word embeddings with the latent code z) as in Hu et al.'s [187]. We append the desired auxiliary input to each step of the decoder in one hot encoding. Thus, for each attribute code in s, we create a separate discriminator to assess the degree to which the generated samples match the desired attributes and motivate the generator to produce better results. The most frequent and salient words within the item set are stance-related in each stance subset [352] e.g. uncertainty, might, probably.

In particular, a decoder GRU (or generator) receives different inputs at every time step: (1) the latent representation for the claim z, (2) different auxiliary input for each warrant generation and the output of master agent and (3) the general context vector. For each decoder, we also provide a representation for the auxiliary information, aux1 for personality and subjectivity, aux2 for the keyword, aux3 for the topic, aux 4 for the target parse tree so at every time step the decoder computes Intuitively; we want the decoder to focus on portions of that correspond with the current time step. As such, we encode the claim using a (unidirectional) GRU and compute z_t with an attention weighted average of the GRU's encoded states at every time step. This attention mechanism is conditioned on the decoder's previous hidden state h_{t-1}.

Constituency and Dependency with Attention Bi-GRU-CNN are used to obtain additional syntactic information. We use the spacy library to extract the text's constituency and dependency structure to preserve the original claim style while generating a new warrant. Thus, the decoder considers the claim's syntactic features at each time step of the decoding process. Additionally, given a sentence and a target syntactic form, we represent the target warrant zt (e.g., a constituency parse). Incorporating the target constituency parse inputs to the decoder generates a warrant with the desired syntax, as in equation 6.59:

Equation 6.59 $s_t = GRU_d\left(s_{t-1}, \left[h_i^{cl}{}_t, E(Y_{t-1)}\,; z; c_{t-1;}^*\,\right]\right)$

The probability p of decoding each word is computed as in equation 6.60:

Equation 6.60 $P = SoftMax\,(W_v\, tanh(s_t;\ [c_{t-1}^*\ ;\ cl_t]; z) + b_v)$

Where st is the state at the current time, z is the claim latent variable, c_{t-1}^* is the master agent output, $cl_t$is auxiliary information, Wv is weight parameters, and bv is bias term. The generator component is GAN which uses a neural network to fool another neural network (the discriminator). It takes the improved vector that has been produced by multiple agents (context vectors), which is efficient and produces a better vector to be decoded by GRU.

Discriminator and Rewarder: The discriminator is an MLP with a SoftMax layer that distinguishes generated tokens from real tokens to maximise the multi-agent model's total expected future reward. By observing the discriminator and rewarder losses, the RL agent determines the optimal input GAN. We use CNN for Discriminator to discriminate between fake and real arguments. The discriminator is the similarity between generated and factual arguments' representations. Sigmoid (f) is the signal from the discriminator Our model will use the GRU Autoencoder to determine whether a data sample is fake or real. Autoencoders are feed-forward neural networks trained to learn the most salient features similar to those found in real news. The function f's hidden output is reconstructed using function g reconstruction, which preserves the variable distribution. The term "error backpropagation" refers to the sum of the distances between real and fake points, which is significantly greater for false sequences, reconstruction error [188]. We reward the generated warrants by bleu metrics. The rewarder preserves the quality of warrants and acts as generation guidance.

For decoding the encoded information from the generator, switching the pointer generator network (generators conditioned) will be applied. Pointer Generator will be used due to its ability to deal with Out-of-Vocabulary (OOV). A switching pointer generator network to generate the sequence of tokens Y1 … Yt (warrants) is used in our decoder work as it proves competitive results [401], [404], [405]. We evaluate our models with the other two metrics used in Park et al.'s model [339].

6.3. Experiments and Results

Experiments are conducted to assess our model's performance in terms of both the quality and diversity of automatic evaluation metrics.

6.3.1. Implementation Details

We implement our model using Keras and a pre-trained 300-dimensional Glove word Embedding [146]. The encoder employs 300-dimensional hidden states, while the decoder employs 300-dimensional hidden states. We use the Adam optimizer [406], with both the encoder and decoder set to a maximum of 50 tokens and the batch size set to 32. The hyperparameter values used in a model have a significant impact on its performance. We tune hyperparameters to achieve a more robust and generalised mode. We create our implementation of an algorithm by determining the optimal hyperparameter values for a given task and dataset. We divide the available data into training and testing subsets, then repetition of optimization loop until a condition is met and finally, we compare all metric values enables you to select the hyperparameter set that produces the optimal metric value.

6.3.2. Dataset

We conduct experiments using data from the ARCC [365] repository, which is annotated in such a way that serves our work. Habernal et al. [365] developed the ARCC to discover warrants. It contains 188 debate topics for the argument reasoning comprehension task as in the following example [365]:

"Reason: Cooperating with Russia on terrorism ignores Russia's overall objectives.
Claim: Russia cannot be a partner.
AW adversarial warrant: Russia has the same objectives of the US.
W warrant: Russia has the opposite objectives of the US.",

Whereas our model is intended to generate warrant W for why the evidence implies the claim. The data set contains 1,970 rows and is divided into three groups: the training set (1,210 rows), the development set (316 rows), and the test set (444 rows). As illustrated in table 6.1, in addition to the claim and evidence, we refer to the primary textual source because the model requires additional inferences as knowledge from the relevant data. The data is contained in a tsv file that has a list of all URLs as well as a list of topics linked to these articles, from which we could recover and obtain the complete relevant articles for warrant information.

6.3.3. Evaluation Methods

The quality and diversity of generated text, are widely used in Park et al.'s [339] text generation task model and will be used in our evaluation [339]. Quality metrics include BLEU-1/2 and Embedding Average/Greedy/Extreme, while diversity metrics include Dist-1/2 and Dist-1/2-within the generated warrant for each. Given that evidence used to substantiate a claim may cover a variety of aspects of an argumentative topic, the diversity and quality of generated text should be evaluated to determine the breadth and variety of word usage in writing, as well as the vocabulary richness and n-gram precision desired in conversational topics. We compare how close the generated warrant with the top ranked warrant provided by the dataset.

- BLEU-1/2: measures N-gram precision of the generated text to multiple target arguments references [407]
- Embedding Average/Greedy/Extreme: measures the semantic similarity between hypothesis and references, using a semantic representation by word embedding [408]
- Dist-1/2: computes the percentage of unique unigrams/bigrams within a sentence to measure the diversity among multiple generated texts [408]
- Dist-1/2-within [339], propose a simple metric to calculate the sum of the numbers of unique N-grams for each result that does not occur in other results) / (The sum of all generated numbers of unigrams/bigrams).

For implicit reasoning, current approaches either locate multiple warrants from an existing structured corpus of arguments via similarity search [355], [396] or incorporate them to improve the performance of evidence detection [355]. While Singh et al. [355] commissioned two annotators to assess the quality of warrants located from the ARCC (ARC Corpus) dataset to various datasets. The proposed method is based on a publicly available dataset ARCC, which stands for Argument Reasoning Comprehension Corpus from News Comments [187], which was built for the 2018 SemEval task [366] by Habernal et al. [365].

6.3.4. Evaluation and Results

To evaluate the quality of our warrant generator and the score of their quality, we use automatic evaluation methods, same to Park et al.'s model [339] evaluation metric, as in table 6.4 for quality and table 6.5 for the diversity. We conduct ablation experiments to demonstrate the effectiveness of reinforcement learning and its associated benefits in terms of generating more enhanced warrants.

Table 6.4: Automatic evaluation results on warrant generation quality in our proposed model warrant generators.

Method	**BLUE-1**	**BLUE-2**	**Embedding Average**	**Embedding Greedy**	**Embedding extreme**
Lexical Chain with Multi-Head Attention (without RL-agent)	0.2019	0.0897	0.7107	0.3989	0.2374
Lexical Chain with Multi-Head Attention controlled by RL-agent (SARAS)	0.2974	0.1084	0.7885	0.5265	0.2944
A multi-column convolutional neural network for why-QA (without RL-agent)	0.2717	0.0807	0.6921	0.5282	0.2404
A multi-column convolutional neural network for why-QA Controlled by RL-agent (SARSA)	0.3205	0.1175	0.7744	0.5817	0.2978
RST (without RL-agent)	0.2153	0.0884	0.6408	0.5578	0.3432
RST controlled by RL-agent (DDQN)	0.3381	0.1192	0.7822	0.6168	0.3828
RST-Multi-head attention generator (without RL-agent)	0.3427	0.1069	0.7439	0.5997	0.3834
RST-Multi-head attention generator controlled by RL-agent (DDQN)	0.3749	0.1205	0.7943	0.6227	0.4436
Fine-tune BART on ARCC without adding external knowledge	0.3946	0.1311	0.8083	0.6415	0.4632
Fine-tune BART on ARCC with adding external knowledge	0.4226	0.1468	0.8128	0.6605	0.4743
Fine-tune BART on ARCC with multi-agent	**0.4296**	**0.1491**	**0.8213**	**0.6736**	**0.4887**

Novel hybrid models for warrant generation are proposed in our work, which combines natural language processing, deep learning, and reinforcement learning techniques. Each model is constructed using a new framework that includes a locator and a generator. To generate warrants, the generator is initially trained using sequence-to-sequence learning. The selector, which is used to identify warrants relevant fragments, is then trained in a variety of environments using supervised or reinforcement learning techniques. The goal of reinforcement learning is to find the best reward function for the expert policy. Finally, the generator is fine-tuned further through reinforcement learning to produce more accurate warrants with a well-trained locator. High prediction success rates have been achieved thanks to the diversity of approaches used in the proposed models.

Table 6.5: Automatic evaluation results on the diversity of warrant generation of our proposed model.

Method	**Dist-1**	**Dist-2**	**Dist-1-within**	**Dist-2-within**
Lexical Chain with Multi-Head Attention (without RL-agent)	0.0816	0.0955	0.1993	0.2153
Lexical Chain with Multi-Head Attention controlled by RL-agent (SARAS)	0.1266	0.1225	0.2454	0.2881
A multi-column convolutional neural network for why-QA (without RL-agent)	0.1182	0.2265	0.3103	0.3244
A multi-column convolutional neural network for why-QA Controlled by RL-agent (SARSA)	0.1382	0.2963	0.3422	0.3818
RST (without RL-agent)	0.0927	0.2791	0.2695	0.3364
RST controlled by RL-agent (DDQN)	0.1423	0.3210	0.3612	0.4147
RST-Multi-head attention generator (without RL-agent)	0.1102	0.2983	0.3274	0.3908
RST-Multi-head attention generator controlled by RL-agent (DDQN)	0.1528	0.3291	0.3710	0.5007
Fine-tune BART on ARCC without adding external knowledge	0.1638	0.3478	0.3834	0.5218
Fine-tune BART on ARCC with adding external knowledge	0.1735	0.3574	0.3906	0. 5320
Fine-tune BART on ARCC with multi-agent	**0.1829**	**0.3627**	**0.4003**	**0.5389**

Eleven configurations were compared in this work. We will concentrate on the two configurations below: (1) fine-tune BART using ARCC; (2) fine-tune BART using ARCC and then use a multi-agent model. To evaluate our warrant generator's quality and diversity, we employ automated evaluation methods similar to those used to evaluate our model [409], which are the most widely used automated metrics for comparing system output to gold warrants.

In comparison to our previous work [409], table 6.4 and table 6.5 demonstrate that fine-tuning the ARCC corpus significantly improves the results, but in some cases, it is unable to generate plausible warrant. To address this, we leverage a multi-agent model to generate a more diverse and high-quality warrant based on the BART-generated warrant. The evaluation's findings indicate that fine-tuning BART on ARCC with multiple agents results in competitive performance in nearly every metric related to the quality and diversity of generated text.

6.3.5. Discussion and Analysis

Generally, in terms of metric quality as well as diversity, RST-Multi-head attention generator controlled by RL-agent (DDQN) outperforms all baselines on all test datasets, and the numbers are good enough when compared to the closest work Park et al.’s model [339].

By experimenting with different SARSA and DDQN for each model, we discovered that they make little difference. This means that they reward similar to the generator, resulting in very similar results when changing the RL-agent, for example, from SARSA to DDQN and vice versa. We use reinforcement learning in our models to generate more interesting and coherent warrants focusing on the context of claim and evidence reason. The experiments in tables 6.4 and 6.5 demonstrate that automated diversity and quality metrics produce scores that are significantly higher than the baseline (without Reinforcement Learning). The effectiveness of reinforcement learning, which involves the agent performing an action and being rewarded, is demonstrated by the promising outcomes obtained as a result of the reward used to guide the generator. The best performance is obtained when the RST-based algorithm is combined with multi-head attention for warrant generation enhanced by RL-agent.

According to Al-Khawaldeh et al. [396], the RST-based algorithm for filtering a warrant for a claim trained using DDQN has the highest f-score because it assists in detecting the relationship between clauses. This model can benefit from text organisation by dividing it into sub-clauses, either as a nucleus or a satellite, after the semantic structure is parsed using RST. Since RST is useful for determining the structure and relationship of arguments, this model's performance is enhanced. The more fundamental relationships are interpretation, justification, confirmation, illustration, result, explanation, evidence, foundation, and condition.

Causal relationships between two events establish common causes that support the initial event, assisting in causal inference. Given that a warrant justifies the claim based on the evidence, it improves the model's ability to capture the text fragment that supports the evidence. As a result, we investigated that using a multi-column convolutional neural network for the why-QA model proposed by Oh et al. [385], dealing with warrant generation as Why-question answering (why-QA) that retrieves the warrant as to the answer to a relevant document (evidence) and automatically recognises causalities is extremely practical.

Along with the primary role of the lexical chain, we use the strongest chain as an auxiliary input to select significant sentences. Extracting the highest score (sequence of related words) as an auxiliary input to the model enables the model to pay more attention to the most informative words in the evidence while preserving the main content. In other words, the most robust chain reflects the evidence's central theme. They are extracting the chains of evidence articles to summarise and reduce the data. For Multi-Head Attention CNN-Bi-LSTM, individual attention heads capture more linguistically interpretable representations: syntactic and semantic relations that the encoder finally concatenates to attend to data from distinct representation subspaces. Local and global features are detected using the CNN-Bi-LSTM combination.

The RST-based algorithm, when combined with multi-head attention, outperforms the first model for warrant generation, while Model 3 is the best performer. The primary objective of our work in utilising RST is to return the appropriate warrant from the retrieved evidence in light of the claim. The input that justifies detection is the claim's "bag of words" and relevant evidence. The RST-based method improves the warrant generation, compared to Multi-Head Hierarchical Attention CNN-Bi-LSTM combined with the most robust chain evidence and causality attention.

In this work, we begin by filtration warrants using an RST-based method and then use Multi-Head Hierarchical Attention as a generator controlled by DDQN. In comparison to the other three models, the RST-Multi-head attention generator controlled by RL-agent (DDQN) model produces the highest-quality warrants based on diversity and quality metrics in addition to the f-score measure.

To locate the relevant information of the warrant associated with a particular claim and evidence, it is necessary to determine the context of that claim within the evidence. The RST connection is used to denote which sections of the text contain the warrant (that could be implicit or explicit). A critical property of an RST analysis in RST combined with the Multi-Head -Attention Mechanism model is that RST parses unstructured text into clauses with rhetorical relations, nucleus or satellite, as in the example below. The warrant is connected to the claim in this example via an explanation relation (as a result) in figure 6.3.

To filter warrant using RST, we must first identify text units (spans) within the evidence and then determine their relationships (rhetorical relations that hold between them). Certain rhetorical relations contain cues that connect these spans; for example, the relation result contains a "so," the relation evidence connects the claim with the candidate warrant as a cause-effect relationship, the nucleus is the claim, and information aimed at increasing belief in the claim is considered a warrant in our work. DDQN requires both encoder and decoder to have an informative representation of internal states in the form of hidden vectors. The DDQN learns how to determine which action (e.g., word) to choose from a list to modify the current decoded sequence in the long run. It approximates the Q-value function by updating its Q-values through actions and rewards, selecting the action with the highest Q-value in the outputs.

We observe that when external knowledge is combined with the stated claim and evidence via a promising feature for guiding BART finetuning, the warrant generated is more plausible than those generated by BART fine-tuned on ARCC alone.

Our experiments indicate that more diverse and higher-quality warrants are obtained by encoding sufficient background information from multiple BART-based generated warrants, as opposed to using only one of the finetuned models. Finally, for warrant generation in the third model, it is necessary to model the argumentative context in conjunction with common sense that is already from BART in order to generate a valid warrant that does not violate well-known facts about the world.

We investigated how multi-agent deep reinforcement learning can benefit from the presence of warrants generated by the BART model in the environment to achieve optimal performance. By incorporating model-based auxiliary knowledge and modelling the information of other agents, we can train agents to generate more diverse and high-quality warrants.

6.4. Summary

The warrant element of the Toulmin model is critical for fact checking and assessing the strength of an argument. As implicit information, warrants justify the arguments and explain why the evidence supports the claim. Despite the critical role warrants play in facilitating argument comprehension, the fact that most works aim to select the best warrant from existing structured data and labelled data is scarce presents a fact checking challenge, particularly when the evidence is insufficient, or the conclusion is not inferred or generated well based on the evidence. Additionally, deep learning methods for false information detection face a significant bottleneck due to their training requirement of a large amount of labelled data. Manually annotating data, on the other hand, is a time-consuming and laborious process. Thus, we examined the extent to which warrants can be generated using unstructured data obtained from their premises.

We propose various Deep Learning models for Toulmin Argument warrant generation in this chapter. We demonstrated the performance of each of these models and the benefit of combining them with a reinforcement learning agent to improve generation and inference accuracy. Our investigations confirm that it is necessary to combine our model with auxiliary data such as the topic and sentiment. Incorporating a reinforcement learning agent enables the generator to receive rapid and robust training for decoding sequential text successfully. We generate warrants using RST and attention mechanism and obtain the best results on the ARCC dataset [365].

We present an end-to-end approach to developing a new model for automatically generating warrants based on a claim supported by evidence. We demonstrate how utilising pre-trained language can significantly improve the performance of a state-of-the-art generative language model used for warrant generation. Finally, we enhance the generation process that uses a multi-agent model to generate an enhanced warrant that outperforms all existing baselines in terms of diversity and quality automatic metrics.

Chapter Seven: **Factual Claim Generation**

7.1. Introduction

Due to the difficulty of explaining the internal workings of deep neural networks, which are frequently used for fact checking models, explainable fact checking models are necessary. Explainable fact checking models assist end-users in comprehending why the model predicted a given label, such as a false claim, and whether or not the classification can be relied upon.

In the previous chapters, we discussed how a warrant can be a plausible explanation for fact checking models, explaining why the evidence implies the claim. Because it is typically implicit and may be filled in or inferred by users, its absence limits the model's performance. Thus, having an explicit warrant could result in improved performance. A primary requirement for models of warrant generation based on deep learning is that they require externally relevant articles focusing on the claim's target. Such information may necessitate a lengthy process, such as retrieving Web pages via search engines and then automatically extracting warrant-relevant sentences from these pages in order to generate warrants as credible explanations for the claim based on the evidence.

In this chapter, we introduce a method for generating factual claims as an alternative solution for explainable fact checking. Our mission is to generate factual claim in order to make them more evidence-based and while fact checking merely classifies the claim's veracity. Also, we examine the applicability of repurposing available data for fact checking due to a scarcity of corrected claims data. RQ4 "To what extent does the generation of factual claims, as explanation for the reason behind the decision, affect false information detection performance?" is discussed in this chapter.

Models that predict labels for fact checking continue to identify a large proportion of false claims without providing any context. The decision-making process should be clarified by generating corrections. The idea is that the proposed check an initial claim and, if necessary, correct it to demonstrate that the model does not decide on a false label without interpretation to reduce decision-making ambiguity. We hypothesise that the task of correcting false claims can elicit information about why people believe the claim is true and has been legitimised. Correcting false information benefits the individuals who may be harmed by it. To accomplish this, we propose two models: the generator model for creating factual claims and the modifier model for modifying the false claim to make it more precise by modifying the claim's misleading information. The goal of developing two models is to see which is more feasible for improving fact checking model performance: editing a claim to be correct based on evidence and then checking the correction claim against the claim to be checked (e.g., user claim) or generating a claim based on evidence and checking it against the user claim regardless of what the user claim is. The generator model's objective is to re-assess the claim's truthfulness using relevant data and generate new claims based on the claim's primary aspect and supporting evidence. We determine whether or not the new assertion is true. A correction model based on sequence operations is proposed for detecting false information and modifying it to make a factual claim. Consider the following illustration. For instance, in the PERSPECTRUM dataset [3], for example:

The user claim must be verified: "animals have no interest or rationality".
Evidence: "The principle of equality advocates equal consideration, so it still allows for different treatment and different rights".
The correct generated claim based on the evidence of our proposed model: "Animals should have lawful rights"
The editable user claim based on the evidence of our proposed model: "animals have interest and rationality".
According to this perspective and it’s the generated correct claim and the editable claim, the claim is false.
Other generated claims by our model: “Animals have a sense of curiosity and reason.”, “Animals should be able to exercise their legal rights.”

7.2. Our Proposed Model Architecture

To ascertain the claim's veracity, we begin by segmenting each sentence into several clauses using sentence-level discourse segmentation and then determining whether a clause is related to the claim or not using cosine similarity. The model is fed the most closely related clauses as evidence input. First, we compare the evidence clauses to the claim; if the claim has a high correlation with the evidence clauses, it is probably correct. Otherwise, proceed to the Hierarchical Reinforcement Learning (HRL). The general architecture of our factual generator is depicted in figure 7.1, where each claim is verified using the substantiating evidence clauses.

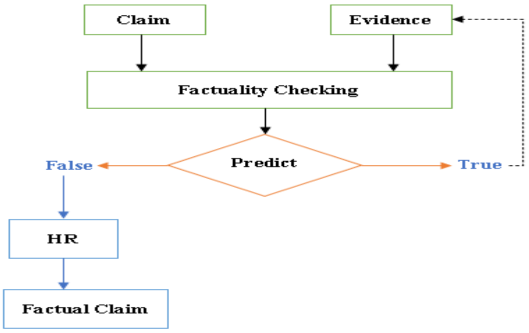


Figure 7.1: The proposed factual claim generator model

7.3. Factuality Checking

We propose a novel Siamese network architecture that combines a multi-channel LSTM, a GRU, and a CNN. This model generates vectors of word embeddings for use with the LSTM-GRU-CNN channels model. Then, we combine all of the features from the various channels for both claim and evidence to arrive at a single numeric score. The Manhattan distance is used to quantify the similarity between two objects. The multi-channel approach is capable of capturing both high-level characteristics and long-term dependencies. We use pre-trained Word Vectors, Glove Embeddings, and Elmo for lexical embedding.

To generate encoded sequences, we use a variety of deep learning models, including Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Long Gated Recurrent Unit (BiGRU), and Convolutional Neural Networks (CNN). To represent the word series and learn long-term dependencies using BiLSTM, we obtain the summarised representation from both the input and the neural network output. They are used to encrypt the sentences. Improved version of BiLSTM for rapid text acquisition in class and subsequent easy access to text functionality.

We use CNN to extract the most pertinent and important information and take advantage of significant and local features. While CNN is unable to capture global features and long-distance events, it is more efficient at training. These characteristics will be combined to create a vector that represents both the claim and the evidence. Manhattan distance is used to determine the credibility of a claim concerning evidence [410]. Our recommended factual checker model is summarised in figure 7.2.

7.4. Generator Model: Factual Claim Generator-Based Hierarchical Reinforcement Learning Approach

For non-factual assertions, all related evidence clauses {c1, c2 ... Cn} is sent to the HRL; word and clause level claim attention is applied by the high-level policy to select the more claim-relevant clauses. All appropriate clauses {c_1, c_2 .. The medium-level policy will be sent to c_{n-m}}, where deep communication agents are implemented to encode these clauses, helping to decide the next sub-goal (copy or generate). The low-level policy has a role in implementing the actions to generate the word sequence (choosing words to create the factual claim). Figure 7.3 illustrates the HRL model for the production of a factual claim.

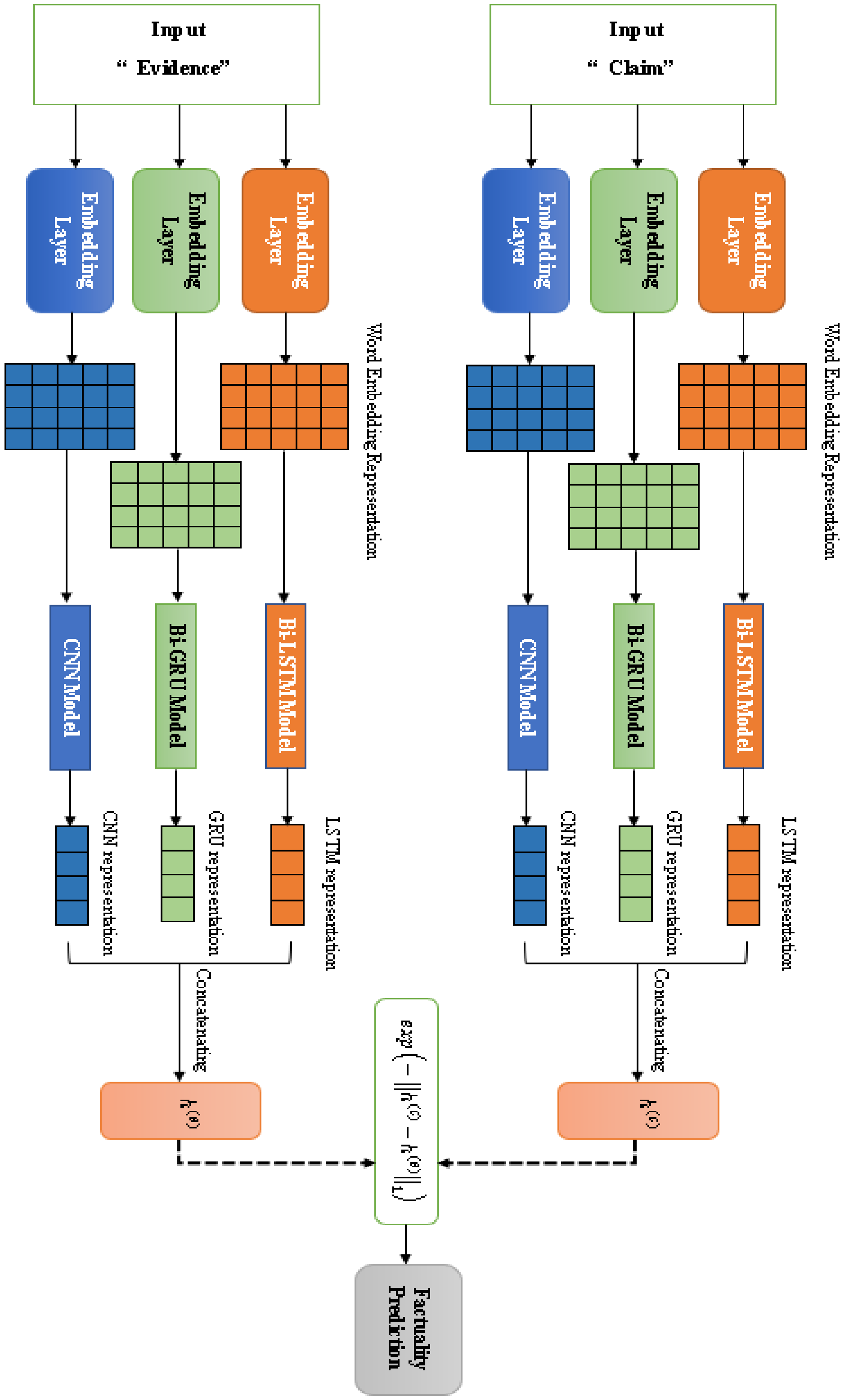


Figure 7.2: Factuality checker model

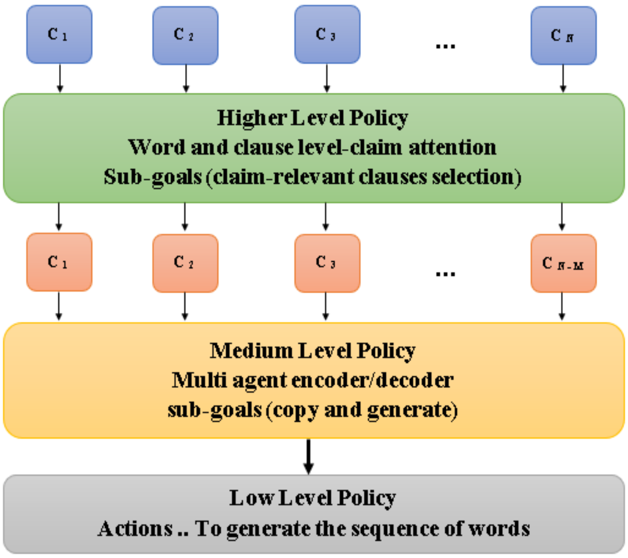

C 1
C 2
C 3
...
C N
Higher Level Policy
Word and clause level-claim attention
Sub-goals (claim-relevant clauses selection)
C 1
C 2
C 3
...
C N-M
Medium Level Policy
Multi agent encoder/decoder
sub-goals (copy and generate)
Low Level Policy
Actions .. To generate the sequence of words

Figure 7.3: Hierarchical Reinforcement Learning (HRL)

7.4.1. Higher Level Policy

For claim-relevant clauses, high-level policy adopts the hierarchical attention mechanism, word-level and clause-level attention networks, to select informative words and clauses relevant to a specific claim.

In a word-level claim attention network, the word encoding layer concatenates claim representation to each word embedding and then summarises information by bi-directional GRU. For each evidence, Bi-GRU (Gated Recurrent Units) will be used to encode the word information in each clause from forward and backward direction, as in equations 7.1-7.3:

Equation 7. 1 $\overrightarrow{h}_{ij} = \overrightarrow{GRU}_{(\widehat{w}_{ij})};\ \ i \in [1,\ C],\ j \in [1, N_i]$

Equation 7. 2 $\overleftarrow{h}_{ij} = \overleftarrow{GRU}_{(\widehat{w}_{ij})};\ \ i \in [1,\ C],\ j \in [N_i, 1]$

Equation 7. 3 $h_{ij} = \overrightarrow{h}_{ij} \oplus \overleftarrow{h}_{ij}$

The Word attention layer focuses on the terms that are important to the meaning of the clause concerning the claim, producing clause vectors. Attention mechanism will be implemented to concentrate on those words in the evidence clause concerning a specific claim CP and combine the representation of all of them to form a clause vector of evidence as in equations 7.4-7.6:

Equation 7. 4 $u_{ij} = tanh\left(W_w \cdot \left[h_{ij}; CP\right] + b_w\right)$

Equation 7. 5 $a_{ij} = softmax\left(u_{ij}\right) = \frac{exp(u_{ij})}{\sum_{t=1}^{N} exp(u_{it})}$

Equation 7. 6 $c_i = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij}$

Clause encoding layer applies Bi-directional GRU to capture the context clause representations. BI-GRU obtains the contextual information of each clause as in equations 7.7-7.9:

Equation 7. 7 $\vec{h}_i = \overrightarrow{GRU}_{(c_i)}; \quad i \in [1,\ C]$

Equation 7. 8 $\overleftarrow{h}_i = \overleftarrow{GRU}_{(c_i)}; \quad i \in [C, 1]$

Equation 7. 9 $h_i = \vec{h}_i \oplus \overleftarrow{h}_i$

After that, in the clause attention Layer, the attention mechanism computes the attention weight between each claim-clause representation to produce contextual information conditioned on the claim representation. For example, the attention weight between each clause and the representation of a specific claim will be computed as in equations 7.10 and 7.11:

Equation 7. 10 $m_i = tanh(W_c \cdot [h_i; CP] + b_c)$

Equation 7. 11 $a_i = softmax(m_i) = \frac{exp(m_i)}{\sum_{t=1}^{C} exp(m_t)}$

In this policy, to select claim-relevant clauses, conditional probability is used. The selected clauses are sent to the middle-level policy, where the multi-agent encoder is used to generate hidden states for the evidence clauses considering the claimed interest.

7.4.2. Middle-Level Policy: Multi-Agent Encoder

The context and states from the environment are used to create all possible sub-goals (to copy or to generate), which should be achieved by the lower agent policy to select a series of actions (words) and produce a new sequence of words. We depend on relevant clauses segments as input and then apply a stack of deep learning models: CNN and MaxPooling layer+ GRU. We use the message sharing mechanism to help other agents' encoders to generate better contextual information conditioned upon the messages received from other agents. For the multi-agent encoder, we use equations 7.12-7.17 where message passing is applied:

Equation 7. 12 $\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)} = bGRU\left(e_i, \vec{h}_{i-1}^{(1)}, \overleftarrow{h}_{i+1}^{(1)}\right)$

Equation 7. 13 $h_i^{(1)} = W_1\left[\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)}\right]$

Equation 7. 14 $\vec{\mathrm{h}}_{\mathrm{i}}^{(\mathrm{l}+1)}, \overleftarrow{\mathrm{h}}_{\mathrm{i}}^{(\mathrm{l}+1)} = \mathrm{bGRU}\begin{pmatrix}\mathrm{fun}\left(\mathrm{h}_{\mathrm{i}}^{(\mathrm{l})}, \mathrm{mes}^{(\mathrm{l})}\right), \\ \vec{\mathrm{h}}_{\mathrm{i}-1}^{(\mathrm{l}+1)}, \overleftarrow{\mathrm{h}}_{\mathrm{i}+1}^{(\mathrm{l}+1)}\end{pmatrix}$

Equation 7. 15 $h_i^{(l+1)} = W_2\left[\vec{h}_i^{(l+1)}, \overleftarrow{h}_i^{(l+1)}\right]$

Equation 7. 16 $mes^{(l)} = \frac{1}{N-1}\sum_{n\neq a} h_{n,I}^{(l)}$

Equation 7. 17 $fun = v_1^T \tanh\left(W_3 h_i^{(l)} + W_4 mes^{(l)}\right)$

e_i is word embedding, $h_i^{(1)}$ is the concatenation for both directions for hidden states before considering other agent information, mes is the encoded information from other clauses, and fun is the score function.

Decoder with Claim and Evidence Attentions: Inspired by Celikyilmaz et al. [411], to guide the decoder's focus on the claim concentrated aspect, the decoder calculates the attention weights for every word in the claim and evidence calculated by Claim attention and Evidence attentions, respectively.

Claim attention: according to equations 7.18-7.21:

Equation 7. 18 $s_t = GRU_d\left(s_{t-1}, \left[h_i^{cl}{}_t, E(\mathcal{Y}_{t-1}); c_{t-1}^*\right]\right)$

Equation 7. 19 $a_{t,i}^{cl} = v_{cl} \cdot \tanh\left(W_{cl} s_t + U_{cl} h_i^{cl}\right)$

Equation 7. 20 $\alpha_{t,i}^{cl} = \frac{exp(a_{t,i}^{cl})}{\sum_{i=1}^{|cl|} exp(a_{t,i}^{cl})}$

Equation 7. 21 $cl_t = \sum_{i=1}^{|cl|} \alpha_{t,i}^{cl} h_i^{cl} \cdot$

Evidence attentions (word attention distribution): according to equations 7.22 and 7.23:

Equation 7. 22 $\mathrm{a}_{t,j}^{d} = v_d \cdot \tanh\left(W_d s_t + U_d h_j^l + Z cl_t\right)$

Equation 7. 23 $a_{t,j}^{d} = \frac{exp\left(a_{t,j}^{d}\right)}{\sum_{j=1}^{|w|} exp\left(a_{t,j}^{d}\right)}$

$d_a^t = \sum_i a_{t,j}^d h_{a,i}^{(l)}$ For each clause (by each agent)

Agent Attention: The last hidden state from each agent is sent to the decoder to compute the global agent attention as in equations 7.24-7.27:

Equation 7. 24 $agent^t = (\tanh(W_7 d^t + W_8 s_t + b_2))$

Equation 7. 25 $agent^{att} = softmax(agent^t)$

Equation 7. 26 $c_t^{new} = \sum_a agent_a^{att} d_a^t$

Equation 7. 27 $P_{subgoal} = sigmoid\left(w_g * (s_t + \text{yt-1} + c_t^{new}) + b_g\right)$

st is a state that is computed by the decoder by attending to relevant input context provided by the agents, *yt*-1 is the previous target word, c_t^{new} is the agent context vector.

7.4.3. Low-Level Policy

After receiving sub-goals from the middle-level policy, low-level policy performs necessary actions to achieve the specified goal (selecting words), following equations 7.28 and 7.29:

Equation 7. 28 $P^{voc}(w_t) = softmax\left(MLP([s_t, c_t^{new}])\right)$

$P^{voc} is$ Vocabulary distribution

Equation 7. 29 $P_{action} = P_{subgoal(generate)} * P^{voc} + P_{subgoal(copy)} * \sum_{i:w_i=w} a_{it}$

$y_{t-1}$is the embedding vector of the previously generated word.
The final evidence hidden states will initialise the first state of the GRU in the decoder. If the word is an out-of-vocabulary (OOV) word, then Pvocab (w) is zero; similarly, if w does not appear in the source document, then $\sum_{i:w_i=w} a_{it}$ is zero.

Multi Rewards: We apply a rewarder function to compute the new claim's factuality using entailment and semantic similarity metrics to find a policy $\pi *$ that maximises the reward for each visited state s and action a, as in equation 7.30:

Equation 7. 30 $\pi_j^{*i}(s) = \arg\max Q_j^{*i}(s, a)$

For high-level policy, the cumulative reward is calculated between the claim embedding and the selected candidate clause, using cosine similarity as in equation 7.31:

Equation 7. 31 $r_i^h = \leftthreetimes_1 \sum_{t=i}^n \gamma^{t-i} \log cos(v_a, \hat{v}_t)$

We calculated the entailment probability score between the evidence (as a premise) for low-level policy and generated a factual claim (as a hypothesis). We apply the Entailment Corrected Reward in Pasunuru & Bansal [412].

7.5. Modifier Model: Sequence Operation Based Hierarchical Reinforcement Learning Approach

We propose a hierarchical reinforcement learning approach for claim factuality prediction and adjusting it if its factuality is incorrect where different polices are applied. The main idea of the proposed approach is to perform a factuality prediction. In particular, our approach employs a high-level policy to select appropriate clauses and a low-level policy to adjust claims that meet the appropriate standards. We have a hierarchical reinforcement learning approach with a high-level clause selector, a low-level claim adjustor, and a fact predictor to provide ongoing rewards to guide both the clause selector and claim adjuster. In the absence of a clause, the policy decides whether the evidence article mentions the claim. If the statement is relevant and not noisy, the high-level policy picks the clause and sends it to the low-level policy, which aggregates the clauses that way and considers them for claim adjusting by selecting one action at each time to change misleading information to be true. After preliminary action is taken, the claim is checked to provide a reward to direct the clause selection and adjust the prediction.

7.5.1. High-level Policy: Claim Relevant Clauses Selector

First, a high-level policy is proposed to select claim-relevant clauses and remove irrelevant clauses. **State**: given the clauses of the article and a claim as input, the policy aimed to decide the clam relevant clauses and passed the selected clauses to the low-level policy that took actions to the false claim to alter it be true. Afterword embeddings $e_i$is performed, we use Bi-GRU to get the vector representation of clause $h_s^{(1)} + h_s^{(1)} + h_s^{(2)} + \cdots + h_s^{(n)}$. After getting the hidden state representations of the claim, we perform an average pooling vector $claim^{(l)}$ through equations 7.32-7.35:

Equation 7. 32 $\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)} = bGRU\left(e_i, \vec{h}_{i-1}^{(1)}, \overleftarrow{h}_{i+1}^{(1)}\right)$

Equation 7. 33 $h_i^{(1)} = W_1\left[\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)}\right]$

Equation 7. 34 $claim^{(l)} = \frac{1}{N-1}\sum_j h_j$

Equation 7. 35 $st = h_s^{(1)} + h_s^{(1)} + h_s^{(2)} + \cdots + h_s^{(n)} + claim^{(l)}$

Action: A stochastic policy uses the state information for deciding to select the clause or not. We adopt a logistic function (conditional probability) to decide whether this clause is relevant for a claim or not, as in equation 7.36:

Equation 7. 36 $action = sigmoid(W\ st,\ b)$

Reward: For high-level policy, the high-level cumulative reward is calculated between the claim embedding and the selected candidate clause, using cosine similarity and the signal from fact predictor as in equations 7.37-7.39:

Equation 7. 37 $claim^{(l)} = \frac{1}{N-1}\sum_j h_j$

Equation 7. 38 $clause^{(l)} = \frac{1}{M-1}\sum_i h_j$

Equation 7. 39 $r_i^h = \ \lambda_1 \sum_{t=i}^{n} \gamma^{t-i} \log cos\left(claim^{(l)}, clause^{(l)}\right) + \text{fact predictor}$

λ is weight parameter and γ is the discount factor

7.5.2. Low-level Policy: Claim Adjuster

We model it as an attention-based [413] pointer network, assigning a normalised probability to each position where the misleading information may occur. The clauses representation represents the state $clause^{(l)}$ Furthermore, hi is each position representation of the claim. Action: we adopt an attention-based policy to take action, sequence operation, where "i" denotes each input claim position, as in equations 7.40 and 7.41:

Equation 7. 40 $m_i = tanh\left(W_c \cdot \left[h_i; clause^{(l)}\right] + b_c\right)$

Equation 7. 41 $a_i = softmax(m_i) = \frac{exp(m_i)}{\sum_{t=1}^{C} exp(m_t)}$

M (action l $clause^{(l)}$; i) = softmax (w *hi), hi is the position word where an action should be taken. The actions are inserted, delete or replace the word

Reward: fact predictor, we apply the double-layer attention mechanism Intra-sequence Attention Layer and Inter-sequence Attention Layer to claim contextual features extraction as shown in figure 7.4. We propose a new model, incorporate the claim-relevant clauses with a double-layer attention mechanism: Intra-sequence Attention Layer and Inter-sequence Attention Layer to capture latent correlation features among claim-relevant clauses sequence. Intra-sequence Attention Layer (intra-relation reasoning) and Inter-sequence Attention Layer used to obtain the characteristic representation of the claim-relevant clauses and find the characteristic representation of the claim-relevant clauses, as in equations 7.42-7.45:

Equation 7. 42 $V = BIGRU(clause)$

Equation 7. 43 $V_c = tanh(V)$

Equation 7. 44 $\alpha = softmax(v \cdot V_c{}^{T})$

Equation 7. 45 $ri = tanh(\alpha \cdot V)$

The representation of the claim-relevance clause is according to equations 7.46-7.48:

Equation 7. 46 $V_{cs} = tanh(R_t)$

Equation 7. 47 $\alpha_t = softmax(v_t \cdot V_{ct}{}^{T})$

Equation 7. 48 $r_e = tanh(\alpha_t \cdot R_t)$

R_t *is* clause sequence features, r_e *is* characteristic representation for all claim-relevant clauses. A SoftMax layer performs the final factuality prediction output as the classifier as in equation 7.49:

Equation 7. 49 $out = softmax(v \cdot r_e + b)$

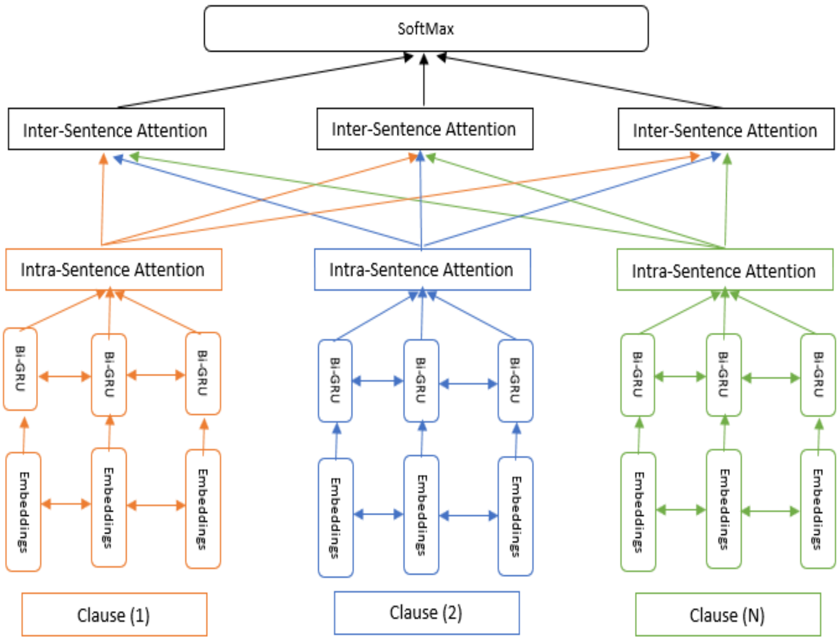


Figure 7.4: Fact predictor model

7.6. Experimental Results

We used the publicly available dataset PERSPECTRUM that was provided by Bahdanau et al. [413]. A collection of a claim, perspective, and evidence statements. The data contains 907 claims, 11,164 perspectives and 8,092 evidence paragraphs. For the factual checking model, to check the claim's factualness, we suppose that it has support from its related perspective, then it should be factual. Thus, comparing the claim to the more concise factual claim generated from its lengthy evidence yields better results than comparing the claim to the lengthy evidence.

7.6.1. Evaluation Model

The factual claim (generated) should contain statements supporting the original perspective of the claim. We offer a bi-GRU Siamese network with attention adaptations. The bi-GRU output should be multiplied by a weight, which again is determined by the claim. Using a BI-GRU based Siamese architecture (it is two networks with the same structure and the same weight, each process one sentence in a pair) to model both claim *m* and perspective *p*; where *hi* is the hidden state of the GRU at time-step i, (or annotation), briefing all the information of the sentence. *c(a,h)* is an annotation attention mechanism that assigns a weight ai to each word annotation, which indicates its importance, and *z* is e the final representation and *y* is the label of the relation between the generated claim and a perspective. (W_1, W_2, …, W_n) is a sequence of words of claim, and (W_1, W_2, W_3, …, W_m) is a sequence of words of perspective. Figure 7.5 shows the information flow through the proposed model used as evaluation model. The predicted labels vs. ground truth are used to calculate the total F1 score.

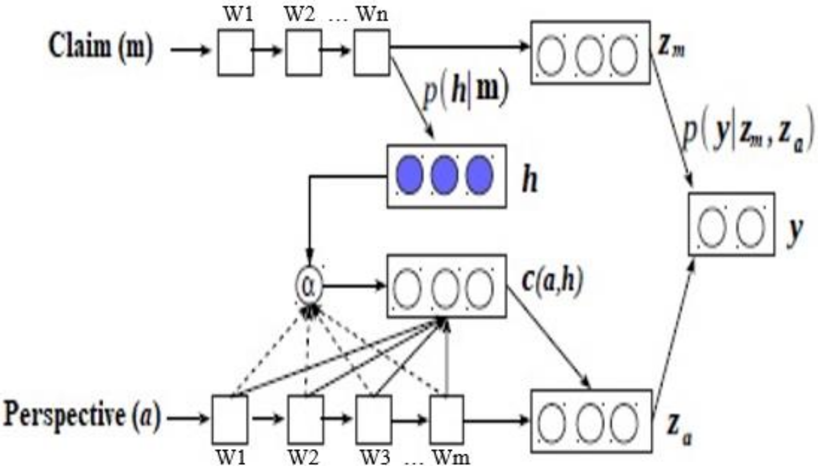


Figure 7.5: The improved factuality checking prediction model [414]

Note that the ROUGE [415] is frequently used to assess the quality of summary generated text, and we utilise it here because the generated text is near to the summarised text. The ROUGE measures the unigram overlap, bigram overlap, and longest common sub-sequence between the predicted and reference. We tried evaluating our approach to the Perspectrum dataset using factor analysis. Table 7.1 shows ROUGE results of claims generated and corrected.

Table 7.1: ROUGE results of claims generated and corrected

The proposed factual claim model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
the generator model	27.41	7.93	25.83
the modifier model	28.50	9.73	27.36

The F-SCORE makes sense of the balance between the generated text's accuracy and the user's evaluation of the text's accuracy. Precision refers to how accurate the model is at duplicating the text from the reference. The recall defines the complexity of the reference text by its frequency of occurrence. For the correct label, we focus on evidence that supports the perspective. That specific perspective entails the claim. Our experiments provide evidence that our system generates factually accurate statements.

Table 7.2: Claim fact checking F-scores after generated and corrected claims

The proposed model	**F-score**
The baseline: Factuality Checker Model as in figure 7.2	73.54%
The baseline results after using generator model	76.84%
The baseline results after using modifier model	78.36%
The improved Factuality checking prediction model as in figure 7.5	79.50%
The improved Factuality checking prediction model results after using generator model	82.94%
The improved factuality checking prediction model after using modifier model	**84.02%**

Based on our Factuality Checking Prediction Model, table 7.2 illustrates the F-score outcomes when prediction labels are compared to ground truth labels. The results are shown demonstrate that modifying the false claim's misleading information is more effective for obtaining a factual claim than generating a new claim from its premises. Our detailed analysis shows that our model modifier performs better than the generator model according to all evaluation methods. To disambiguate the claim, given evidence, the factual claim is checked against the claim, and then the relation is decided. Our methodology generates more concise factual claims, which are then compared to the original claim for fact checking, producing better findings, while the vast evidence used to evaluate the claim comprises a large amount of data that could cause the model to make an inaccurate decision. As results it addressed RQ4 and confirm its hypothesis H4.

7.7. Summary

This work proposes a novel task for supervised learning based on an approach based on objective claims. We develop neural network models that generate correct claims based on contextual information if the original claim is false. We discover that the neural network-based model performs better when the misleading information is modified rather than when a new claim is generated from its premises. We investigated encoding lengthy evidence articles to generate a factual claim for the claim generating model and demonstrated that hierarchical reinforcement learning could improve the generation by automatic evaluation. The analysis demonstrates that this improvement is due to the multi-ability agents covering all pertinent information in the claim and generating a factual claim. The sequence operation-based method combined with hierarchical reinforcement learning (HRL) effectively addresses the non-factual claim problem in the claim modifying model.

Another issue may arise if conflict evidence contributes equally to claim verification in the absence of additional stances from diverse perspectives; thus, it is critical to seek out or generate additional perspectives for use in stance-based claim verification.

Chapter Eight: **Perspectives Generation with Multi-Head Attention Mechanism and Common-Sense Knowledge**

Consideration of multiple viewpoints on a contentious issue is critical for avoiding bias and assisting in the formulation of rational decisions. We observe that the current models impose a constraint on diversity. This is because the conventional attention mechanism is biased toward a single semantic aspect of the claim, whereas the claim may contain multiple semantic aspects. Additionally, disregarding common-sense knowledge may result in generating perspectives that violate known facts about the world. The proposed approach is divided into two stages: the first stage considers multiple semantic aspects, which results in more diverse generated perspectives; the second stage improves the quality of generated perspectives by incorporating common-sense knowledge. We train the model on each stage using reinforcement learning and automated metric scores. The experimental results demonstrate the effectiveness of our proposed model in generating a broader range of perspectives on a contentious subject.

8.1. Introduction

Individuals' assessments of factual truth vary due to their varying levels of subject knowledge and their linguistic abilities. Additionally, the rapid pace, enormous volume, and noise associated with data generated by users with questionable authorship and authenticity result in the emerging claims in a variety of domains, necessitating the consideration of alternative perspectives. It is not always possible to substantiate a claim with an authoritative source, especially when previously unmentioned claims are discovered. Viewing a claim through a singular lens may introduce bias. Without taking into account additional data, relying exclusively on textual information from a single source is likely to result in inaccuracies and bias. To address this issue, it is necessary to critically analyse a claim from multiple perspectives. Regrettably, for rapidly evolving claims and responses, there is a dearth of diverse perspectives on specific claims arising from previously unseen events that are not covered by the databases upon which a retrieval-based system is based.

An argumentative text's objective is to persuade the reader to concur with a particular conclusion. Each argument has a conclusion, supported by one or more premises. By taking into account a diverse range of viewpoints, i.e., perspectives relevant to a given claim for veracity prediction, analysing diverse arguments helps to alleviate the bias problem. To address this issue, the majority of current systems attempt to extract or generate the omitted conclusion from relevant evidence, which may contain a large amount of information about numerous topic aspects as well as additional data that supports or refutes the claim. The diversity is limited to claims supported by facts that can be retrieved only from the datasets.

Also, optimising model prediction via cross-entropy loss alone is insufficient to encourage the model to generate diverse statements [416]. Due to the fact that prediction is made against a single score, at the word level, strict sequence matching between the generated perspective and the ground truth perspective is required.

Additionally, limiting perspectives to a single semantic aspect reduces diversity, as a claim may have multiple aspects. Besides that, even though common-sense knowledge is critical for perspective formation, current systems overlook it such as the state of the art model, Park et al.'s model [339]. While conventional models generate generic responses, the perspectives they generate may contradict common sense. We observe that prior work has a low-performance level, and in this proposed work, we seek to improve the performance, quality, and diversity of a state-of-the-art system in response to a specific claim.

Additionally, as demonstrated by the state of art Park et al.'s model [339], it still performs poorly in terms of quality and diversity on some data sets. Park et al.'s model [339] generates N distinct perspectives and selects the one with the lowest negative log-likelihood NLL for the given reference perspective as the generated perspective. Even though they followed the multi-head attention work [417] and achieved state-of-the-art results in a perspective generation, we believe that multiple automated metrics as a reinforcement learning reward will further improve the approach to controlling perspective generation for optimization and will encourage the model to consider various factors that are necessary to improve the quality a during generation. Additionally, we hypothesise that incorporating common sense makes generated perspectives more plausible and does not violate world facts, which were not considered in the state-of-the-art model. As a result, we begin by introducing a novel technique for generating diverse and high-quality perspectives by focusing on various semantic aspects. The generated perspectives are then supplemented with a collection of common-sense facts. By utilising reinforcement learning, we can combine multiple learning objectives for model training.

This chapter aims to answer the research question, "Can we generate high quality multiple relevant candidate perspectives for a given claim? Our model develops the ability to generate multiple perspectives from its input, which is a claim, an argumentative sentence. The proposed model is trained and evaluated on datasets from Perspectrum [3]. Table 8.1 shows an example of our generator's generation of perspectives alongside a reference from the Perspectrum dataset.

Table 8.1: An example of perspectives generation

Claim: "A government should lessen the economic gap between its rich and poor citizens."	**Perspectives in the Perspectrum dataset**	**Correspondent perspectives generated by our model**
perspective 1	"True individual freedom cannot exist without economic security and independence."	Individual liberty is impossible to achieve without financial security and independence.
perspective 2	"The wealth gap does not allow for equality between the rich and the poor, and so it should be reduced."	Because the wealth gap prevents equality between rich and poor people, it must be narrowed.

In general, the task of generating perspectives relies on supporting evidence and continues to be difficult in terms of quality and diversity. In this work, we propose a novel approach to the task: Given a specific claim about a contentious issue, construct a logical set of perspectives with varying stances. Our primary contributions are as follows:

- Rather than generating a perspective based on a single semantic aspect, as traditional approaches do, we propose a Seq2Seq model with a multi-head attention mechanism that generates diverse perspectives based on the diverse semantic aspects.
- We incorporate common sense knowledge to ensure that the model does not violate known facts about the world and to improve the quality of the generated perspectives.
- We employ a reward function; multi-objective reinforcement learning produces various scores, ensuring that the generated perspectives make appropriate use of the given context and allowing control of the text generation model without relying on a single objective during the decoding process.
- Experiments show that our model outperforms several existing Seq2Seq-based perspectives models on quality and diversity metrics.

8.2. Our Proposed Perspectives Generation Model

In this section, we outline our generation processes and discuss how reinforcement learning can be used to further improve the technique rather than cross-entropy loss. Reinforcement learning is a rapidly growing field of research that involves intelligent agents that learn to reason through Markov Decision Processes [418]. Recently, there has been significant progress that has been used in traditional models in the field of reinforcement learning (DRL) for natural language processing, including relation extraction [419] and reasoning in question answering [420] and generation of paraphrases [421].

To begin, given the claim and some random words replaced with synonyms, the generator is contextualised and more diverse thanks to the wordnet corpus. Then, multiple semantic aspects are extracted, and the claim representation is learned [212], [422] the modified version of the claim with new replaced synonyms words of claim with the semantic extracted aspect are used to guide the generation process to generate candidate perspectives for each semantic aspect. After that, for more information and a better-quality perspective, our models take common sense into account. We used reinforcement learning to enforce diversity, stylistic, and quality constraints on the generated perspective. The overall architecture of our proposed model is shown in figure 8.1:

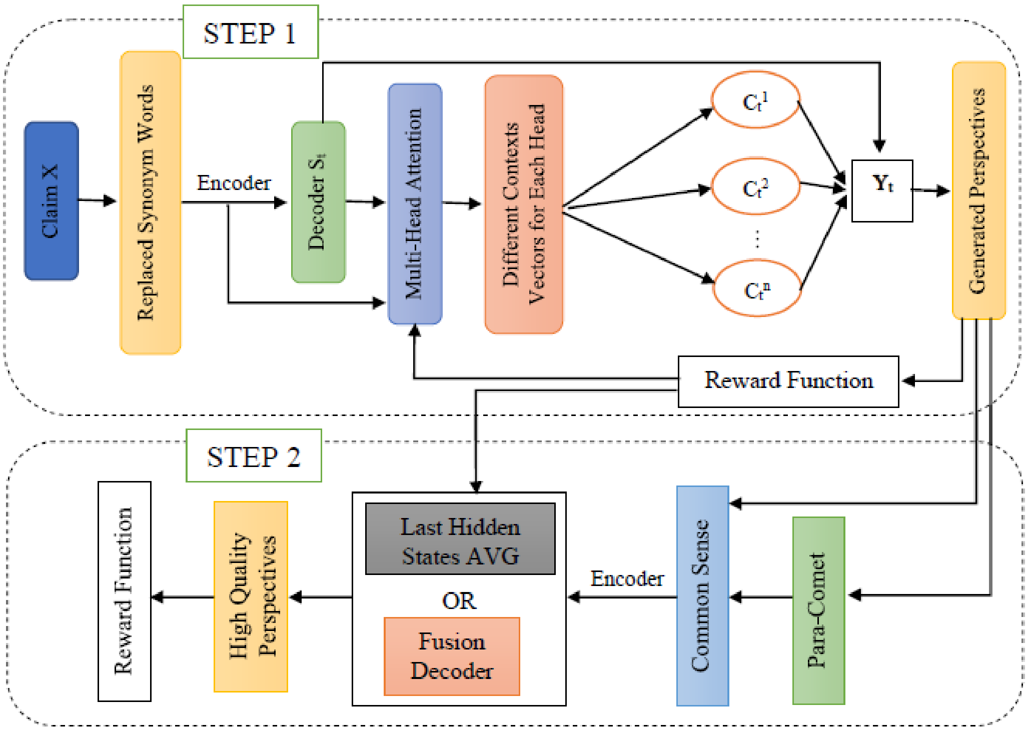


Figure 8.1: Proposed model architecture

8.2.1. First Step: Various Semantic Aspects Considered while Generating Perspectives

8.2.1.1. The Revised Version of the Claim Including Newly Substituted Synonyms

A modified version of a claim is created by replacing specific words in the claim input with synonyms at random. The claim's word sequence is used as input, with specific words being replaced by synonyms at a 60% ratio that fits the semantic aspect context.

8.2.1.2. Multi-head Attention Mechanism

Inspired by the multi-head attention with a Seq2Seq model [417], this work employs a mechanism of multi-head attention to enable the generator to attend to information from different representation subspaces during the generating process, with context vectors obtained via the multi-head attention mechanism focusing on different semantic aspects of the text rather than on a single semantic aspect, as in traditional attention mechanisms, [212], [422].

To generate n distinct perspectives on the claim, n distinct context vectors are created by projecting each state to multiple semantic spaces using various learnable projection matrices as in equation 8.1. The context vector for each head can then be produced by multiplying the encoder's hidden states by a weighted sum for all semantic spaces, the attention process is used to obtain numerous attention probability distributions over the claim words as illustrated in equations 8.2-8.5.

Equation 8.1 $h_j^n = W_t^n . h_j$

Equation 8.2 $a_{t,j}^c = v_{cl} \cdot tanh\left(W_{cl}s_t + U_{cl}h_j^n\right)$

Equation 8.3 $\alpha_{tj}^n = \frac{exp\left(a_{t,j}^n\right)}{\sum_{j=1}^{|c|} exp\left(a_{t,j}^n\right)}$

Equation 8.4 $c_t^n = \sum_{i=1}^{|n|} \alpha_{t,\mathrm{j}}^n h_j^n$

Equation 8.5 $s_t = GRU_{dec}\left(s_{t-1}, \left[c_{t-1}^n, E_{(Y_{t-1})}\right]\right)$

For each semantic space, there is W_t^nlearnable projection matrix, h_i hidden representation for each time-step for the new version of the claim word, $E_{(Y_{t-1})}$ is the previous word embeddings, st is the current state of the decoder at time step t, and $c_t^n$is Context vector for i-th head at a time step where it could be used to generate a i-th perspective that focuses on a particular semantic aspect of the claim. The hidden state of the decoder st at each time t is computed as follow, considering the previous state s_{t-1}, the previous claim context vector c_{t-1}^n and the previous word embeddings.

The probability distribution over the output vocabulary o_t, as equation 8.6 to decide the word which has the highest probability is computed from the context vector ct, and the decoder state st, where $W_g^{(2)}$, $W_g^{(1)}$, $b_g^{(1)}$and $b_g^{(2)}$are learnable parameters:

Equation 8.6 $o_t = W_g^{(2)}\left(W_g^{(1)}[s_t, c_t^n] + b_g^{(1)}\right) + b_g^{(2)}$

$p_t^{pointer}$ is used as a switch to select between [423] (a) copying words from the source text via pointing (copying a word from the input sequence by selection according to the attention distribution) or (b) generating a word from the vocabulary by selecting based on Pv as illustrated in equations 8.7-8.9 where v_{ptr}^{T} and $b_{pointer}$ are learnable parameters.

Equation 8.7 $p_{t}^{pointer} = sigmoid\left(v_{ptr}^{T}\left[s_t, E_{(Y_{t-1})}, \mathrm{c}_{t}^{n}\right] + b_{pointer}\right)$

The generation probability $p_{tj}^{gen} \in [0,1]$ for timestep t is computed as equation 8.8. If p_{tj}^{gen} > 0.5, word is copied from the input determined by the attention distribution where the attention is the highest, else the generator output is used. The probability of generating timestamp t is set to 0. 5 empirically.

Equation 8.8 $p_{tj}^{gen} = \frac{exp(o_{tj})}{\sum_{k} exp(o_{tk})}$

The model then generates distribution Pv over vocabulary. P_v *is* probability distribution over all words in the vocabulary and gives us the final distribution to expect words. It concatenates the output of decoder st as the input of the output projection layer. T, it will show the details of these variables in equation 8.9 where W_v and $b_v$are learnable parameters.

Equation 8.9 $P_v = softmax\left(W_v\left[s_t; E_{(Y_{t-1})}, \mathrm{c}_{t}^{n}\right] + b_v\right)$

8.2.2. Second Step: Conscious of Common-Sense Knowledge to Maintain a Higher-Quality Perspective

Common sense knowledge or world facts are required for the successful completion of a large number of natural language processing tasks [424], [425]. Additional inferences based on common sense knowledge can be formed from a claim accompanied by a modified claim, hence improving the quality of the generated perspectives. To incorporate common-sense inferences into our model we rely on PARA-COMET [426]. As we have two sentences, and a modified claim with random replacements words, we feed this as an input to the trained PARA-COMET model, which generates nine common-sense relations for both sentences "Lemon is sour", for example. For each perspective we have common-sense, for example, [perspective-1, common-sense-1, perspective-2], [perspective-1, common-sense-2, perspective-3], … [perspective-1, common-sense-n-1, perspective-n] and so on for all perspectives. So, to enhance the generated perspective-1, each common-sense relation is encoded. PARA-COMET provides a set of commonsense inferences for the 9 inferential relations for each perspective, based on n-perspectives p1, p2, … pn, that is consistent with the complete narrative. To achieve that, we try two different models that consider the common-sense aiming at enhancing the quality of the generated perspectives. The first enhancer model averages the last hidden states for all related common-sense to i-th perspective and update the context vector. The second enhancer model makes use of a "fusion-in decoder" [427] that is supplemented with common-sense sentences retrieved from external knowledge.

8.2.2.1. Model 1 of the Enhancer: Last of Hidden States AVG

Each agent takes the encoded information $h_i^{(k)}$ from its encoder, which represents a particular generated perspective from the first stage. It considers other agents' information common-sense relations by averaging the last hidden states of other encoders $h_{m,I}^{(k)}$, to produce other important information $v^{(k)}$. An attention vector $f\left(h_i^{(k)}, v^{(k)}\right)$ is produced by considering its encoded feature $h_i^{(k)}$, previous decoder state s^{t-1} and other $v^{(k)}$. Finally, the context vector c_{tj}^{k} is updated based on attention distribution a_{tj}^{k}. Then apply the pointer attention method as in the first stage. The steps are as follows:

- The average of last hidden states for the encoded knowledge common sense relations as in equation 8.10:

Equation 8.10 $$v^{(k)} = \frac{1}{M-1} \sum_{m \neq \alpha} h_{m,I}^{(k)}$$

- Update context vector for each agent, as in equations 8.11, 8.12 and 8.13:

Equation 8.11 $$f\left(h_i^{(k)}, v^{(k)}\right) = v_1^T \tanh\left(W_3 h_i^{(k)} + s^{t-1} + W_4 v^{(k)} + clt\right)$$

Equation 8.12 $$a_{tj}^{k} = \frac{exp(f_{tj})}{\sum_{k=1}^{l} exp(f_{tk})}$$

Equation 8.13 $$c_{tj}^{k} = \sum_{j=1}^{n} a_{tj_i}^{k} h_i$$

W_n are parameters of weights, b_v, v_1^T, and W_3 are learnable parameters

8.2.2.2. Model 2 of the Enhancer: Fusion Decoder

We use the Fusion in Decoder [427] in this enhancer model, a sequence-to-sequence model that accepts as input a previously generated perspective and a set of common-senses from a PARA-COMET [426]. It produces high-quality work that adheres to accepted world facts. Given a perspective with n-1 common senses in support, each common sense is concatenated with the perspective to produce perspectives–common sense contexts. fi = [pi; sj], where fi is encoded separately, but in the decoder the encodings are combined to produce a higher-quality perspective.

8.2.3. Reward Function

As our algorithm attempts to provide varied diverse and high-quality perspectives with distinct stances. So, we use a composite score generated by averaging the specific measures to generate text under various conditions. the average of the individual metrics includes ROUGE, textual entailment, Style **control reward, stance control reward,** diversity and fluency provide a normalised score between 0 and 1. The perspectives are fed to evaluation modules.

8.2.3.1. ROUGE Reward with Reference

To compare the degree to which the generated perspective retains context, it is rewarded using the ROUGE package's primary evaluation metric [415] and the score is then used as a reward. The ROUGE measures the unigram overlap, bigram overlap, and longest common sub-sequence between the predicted and reference [415].

- ROUGE-1: the unigram overlaps describe the overlap of each word between the candidate and reference summaries.
- ROUGE-2: bigram-overlap between the reference summary and the summary to be assessed.
- ROUGE-L: the longest common subsequence between the reference summary and the summary to be assessed.

8.2.3.2. Textual entailment

In terms of supporting perspectives, it should have a higher degree of entailment and a greater reward, whereas attacked perspectives should have a lower degree of entailment and a greater reward. We evaluate our generators using entailment metrics to determine whether the generated perspectives are inferable from (influenced by) the underlying claim. To measure textual entailment, we use a ranking-based loss function to train a model that generates a space embedding for claim contexts and generated perspectives [428].

8.2.3.3. Style Control Reward

We feed the generator with additional style embeddings and can calculate the probability of the output condition based on the style control variable. The rewarder is a convolutional neural network that has been trained to minimise cross-entropy loss in style classification so that the classifier can learn to correctly classify text styles.

8.2.3.4. Stance Module

We can calculate the probability of the output condition based on the stance control variable by feeding the generator with additional stance embeddings. The classifier can learn to correctly classify text stance by using a convolutional neural network that has been trained to minimise cross-entropy loss in stance classification.

8.2.3.5. Fluency

It demonstrates the naturalness of the generated perspectives by measuring the grammatical correctness to increase the probability of the target sentences being used. Each generated perspective is assigned a perplexity level (PPL) by the language model. The less perplexing a perspective is, the more fluent it becomes. We propose to use GPT-2 [429] large-scale pre-trained language models for fluency which is suitable for likelihood-based fluency evaluation and conditional generation.

8.2.3.6. Expression Diversity

Allowing for a wide range of linguistic variations to be captured [430]. Self-BLEU is a tool we use to assess diversity, with a higher score indicating greater diversity. According to BLEU, it calculates the BLEU score for each generated sentence by comparing it to previously generated sentences. By averaging these BLEU scores (for generated sentences), a metric called Self-BLEU is created, with lower values indicating greater diversity.

8.3. Experiments and Results

To evaluate our proposed model, we compare our model to Park et al.'s model [339], which was trained and evaluated on the Perspectrum dataset. 1.970 rows of data divided into three groups: training (1210 rows), development (316 rows), and test (444 rows). Park et al. [339]generate claims in response to a given claim, utilising a diversity penalty to encourage the presentation of diverse perspectives. It utilises a Seq2Seq framework and introduces latent mechanisms on the assumption that each latent mechanism can be associated with a single perspective.

The results in tables 8.2 and 8.3 show that the multi-agent model outperforms baselines in terms of automatic evaluation metrics, diversity, and quality by taking into account various semantic aspects and Common-sense knowledge. We show how leveraging the reinforcement learning reward function improves the perspectives generator model performance of a state-of-the-art model. A multi-agent model, where the decoder network learns from the different semantic aspect vectors during the decoding stage, may capture more realistic arguments than a baseline model. By pooling the common-sense knowledge of multiple agents, the multi-Agent model can capture richer data from multiple perspectives and cover a broader range of issues. Our model can generate high-quality, diverse, and multiple arguments based on the metrics results compared to baseline models. We observe that our model outperforms competitors in all metrics when BLEU score and word embedding-based metrics are used. We achieve the best performance in four metrics (Dist-1, Dist-2, and Dist-1/2-within) for diversity.

Table 8.2: Automatic evaluation results for perspectives generation quality on Perspectrum dataset

Method	**BLUE 1**	**BLUE 2**	**Embedding Average**	**Embedding Greedy**	**Embedding extreme**
Generator: Pointer attention-only one semantic aspect	0.2635	0.0684	0.6838	0.4858	0.2810
ArgDiver [339] as baseline model	0.3268	0.0964	0.8107	0.6002	0.4146
First stage of the Generator: Various semantic aspects of pointer attention	0.3327	0.0919	0.8211	0.6201	0.4139
Second stage of the Generator: Using the average of last states to incorporate common sense	0.3528	0.1027	0.8329	0.6324	0.4476
Second stage of the Generator: Using fusion decoder to incorporate common sense	0.3618	0.1096	0.8514	0.6514	0.4526
Generator rewarded by RL function	**0.3955**	**0.1183**	**0.8801**	**0.6665**	**0.4918**

Table 8.3: Automatic evaluation results on the diversity of perspectives generation on Perspectrum dataset

Method	**Dist-1**	**Dist-2**	**Dist-1-within**	**Dist-2-within**
Generator: Pointer attention-only one semantic aspect	0.1328	0.1983	0.2814	0.4612
ArgDiver [339] as baseline model	0.1585	0.2909	0.3645	0.6134
First stage of the Generator: Various semantic aspects of pointer attention	0.1589	0.2997	0.3729	0.6151
Second stage of the Generator: Using the average of last states to incorporate common sense	0.1603	0.3086	0.4066	0.6272
Second stage of the Generator: Using fusion decoder to incorporate common sense	0.1681	0.3126	0.4182	0.6423
Generator rewarded by RL function	**0.1703**	**0.3208**	**0.4461**	**0.7006**

8.4. An Ablation Study with Automated Evaluation Metric Scores: Quality and Diversity

We investigate our model in depth to develop perspectives for use in an ablation study. The ablation findings are summarised in tables 8.2 and 8.3. We begin with the pointer attention model, which concentrates on a single semantic aspect; the findings indicate that the model achieves the least performance. Our proposed generator has two stages. In the first stage, the generator's effectiveness is evaluated in terms of quality and variety using a metric that considers multiple heads of attention from the model to generate more diverse perspectives. As seen in tables 8.2 and 8.3, on both metrics, it exceeds pointer attention, which concentrates on the same semantic aspect. In the second step, the generator incorporates semantic aspects and common-sense knowledge, and when compared to the outputs of the first stage, the latter outperforms the earlier. This confirms that incorporating world fact makes the generated perspectives more plausible compared to the ones generated in the first stage. The findings of the two independent models are used to compel the generator to combine common sense from various knowledge passages in separate encoders, demonstrating that decoder fusion performs better than taking the average of these knowledge's final hidden states. Experiments and findings reveal that the reinforcement learning-based technique is capable of effectively learning to generate diverse and high-quality paraphrases and greatly increases generation quality when compared to numerous state-of-the-art baselines. Thus, we addressed our fifth research question RQ5 and associated hypothesis H5.

8.5. Summary

This work discusses the generation of perspectives via the employment of several heads of attention to analyse various semantic aspects of a claim. Additionally, we addressed how to maximise the benefits of utilising various common-sense pieces of information. Our approach is capable of generating a variety of high-quality viewpoints on a given claim using a variety of distinct postures, Additionally, we conclude that our reward function advances the state of the art in perspective generation and signals the generation of viewpoints with a specific stance. We examined each step independently in our experiments. We compared our overall strategy to the state-of-the-art approach described in Park et al. [339], using automated evaluation ratings. The results demonstrate that the proposed RL is significantly more performant than a state-of-the-art perspective generative model when considering different semantic aspects and encoding aware common-sense knowledge. In this chapter, we measured the quality of the generated perspectives, but not for their application in fact checking; we'll discuss it in the following chapter.

Chapter Nine: **Unifying False Information Detection Subtasks**

9.1. Introduction

In this chapter, we address three issues identified from the literature that contribute to the failure of veracity prediction systems to achieve acceptable detection performance. The first problem is that stance detection and veracity prediction are separately trained and learned. Even though both stance detection and veracity prediction are positively correlated with joint treatment, current research treats them as distinct tasks, either stance detection [6] or veracity prediction [10]. Because it is not always possible to ground claims in knowledge bases (authoritative sources), particularly for emerging claims, the stances of social media users toward claims can provide indicative clues about their veracity. As a result, the two tasks, stance detection and veracity prediction can be learned concurrently to maximise their utility. The chapter proposes a novel multi-task learning scheme for simultaneously predicting rumour stance and veracity to enhance the performance of a veracity prediction task by leveraging the related task of stance detection, taking into account the strong correlation between claim veracity and the stances expressed in responsive posts.

The second problem is taking stances on lengthy claims with multiple target topics without focusing exclusively on one target topic that receives more attention in response to other claims (replies). For lengthy claims with multiple target topics, as shown in table 9.1, commented on by multiple claims (replies), previous models attempted to detect the general stance without considering the primary or the most concerned target topic. As a result, the stance decision may be incorrect. Therefore, it is essential to extract a specific target topic and examine the stances taken toward the claim in light of this targeted topic. The purpose of extracting the primary common target topic in our proposed model is to eliminate irrelevant and noisy information. Each replay's stance toward this claim is narrowly focused (Each replay is associated with a user who commented on the source). As a result, detecting the target topic and then deriving a target-specific based claim from a lengthy claim and selecting pertinent data assists for stance detection, whereas noisy data contributes less. Another goal of target topic extraction is to classify all claims with associated target topics according to their likelihood of being a specific target topic, then analyse and rank each argument to determine the strongest one. Each claim's target topic is extracted independently. As a result, the target topics with the most similar embeddings to the primary target topic is selected for analysis alongside the target topic. Rumours from reliable sources are weighted heavily in the outcome, whereas rumours from unreliable sources are ignored.

The final problem is that when multiple claims on the same target topic originate from multiple sources may conflict, they are analysed independently, whereas they should all be considered during the claim checking process. Current models tend to be restricted to assessing the veracity of claims (rumours) rather than distinguishing conflicting claims on the same target topic, which results in disagreements when various sources are commenting on the same target topic. Consequently, conflicting statements about the same target topic can be labelled identically, which is illogical because each rumour is independently checked for veracity. Thus, a choice must be made between many conflicting facts about an entity. Given that many statements are made in response to the same thing, we hypothesise that only one of the claims, not the disputes, is credible. Each claim in a natural language argument expresses an opinion about a particular target topic; by incorporating argumentations into the context, the claim can be processed simultaneously with similar target topics, preventing the labelling of competing rumours with the same truth value. The strength of an argument for the proposed system is based on various facts and characteristics derived from either the original content and its account credibility or the replayed content and its account credibility. This is remarkably similar to theories of truth-discovery, which argue that truth is discovered by argumentation.

This chapter focuses on the combination of the stance and veracity detection tasks, and proposed a model for Argument-based Truth Discovery (ATD), to solve the problem with the lengthy rumour with various target topics. In addition, we consider applying truth discovery to integrate information about source credibility which does not account for equal contributions from various sources. The proposed framework incorporates a source credibility metric to compare the strength of arguments to forecast the truth, taking into account both the arguments backed up by supporting sources and the claims rebutted by attacking sources.

The method used in this work predicts both stance and veracity concurrently and establishes a link between bipolar argumentation, in which arguments interact exclusively through attacks and support, and truth discovery techniques. Unlike the models reviewed in section 3.5, our proposed method concludes an article for a specific target topic (the subject of discussion) and learns representative features of stance detection using a different model architecture. Additionally, our goal is to investigate stance classification as a precursor to automatically determining the veracity of a rumour via joint learning to significantly improve these tasks' performance: stance detection and veracity checking.

The remainder of this chapter is structured in the following manner. We present our proposed argumentation-based truth discovery model in Section 9.2. Section 9.3 contains discussion and analysis. Finally, in Section 9.4, we reached a conclusion.

9.2. The Proposed Argumentation-based Truth Discovery Model

Previous research reveals that how people react to rumours can help determine their veracity [285]. The success of multi-task learning in for stance detection task and rumour verification, e.g., by Kochkina et al. [306] and Ma et al. [307], and the observation that people's positions are closely linked to the veracity of the information, prompted us to conduct this study. In contrast to these systems, a new perspective is proposed based on argumentation to consider user trust. To our knowledge, this is the first time that argumentation has been used to model conversations to tackle rumour stance classification and veracity prediction concurrently, to avoid labelling contradictory arguments under the same target topic with the same label. For instance, if both arguments A and B (see below) have the same number of supported (i.e., 3) and refuted (i.e., 2) claims, they are more likely to have the same label, e.g., true claim, despite their conflict.

Argument A: Animal research is the only way to progress at times.
Perspective 1, with support stance: Animal research is only used where other research methods are not suitable.
Perspective 2, with refute stance: Medical breakthroughs can be achieved without doing any scientific or commercial experiment on animals.
Perspective 3, with support stance: Sometimes we have no other choices for Animal research but then to do some animal testing.
Perspective 4, with support stance: Without animal research, we would have fewer products.
Perspective 5, with refute stance: Animal testing does not ensure good results.

Argument B: Animals have a right to live their lives in peace without human interference.
Perspective 1, with support stance: Animal testing significantly harms the animal used.
Perspective 2, with refute stance: Human's rights are a more important consideration than animal rights.
Perspective 3, with refute stance: Innovation often requires the use of animal research.
Perspective 4, with support stance: Medical breakthroughs can be achieved without doing any scientific or commercial experiment on animals.
Perspective 5, with refute stance: Animal testing helps humans.

As a result, the following solution was envisaged: simultaneously considering all arguments relating to the same subject. We propose the Argumentation-Based Truth Discovery Model, abbreviated as ATD. The architecture of ATD is shown in figure 9.1, with the main components as follows:

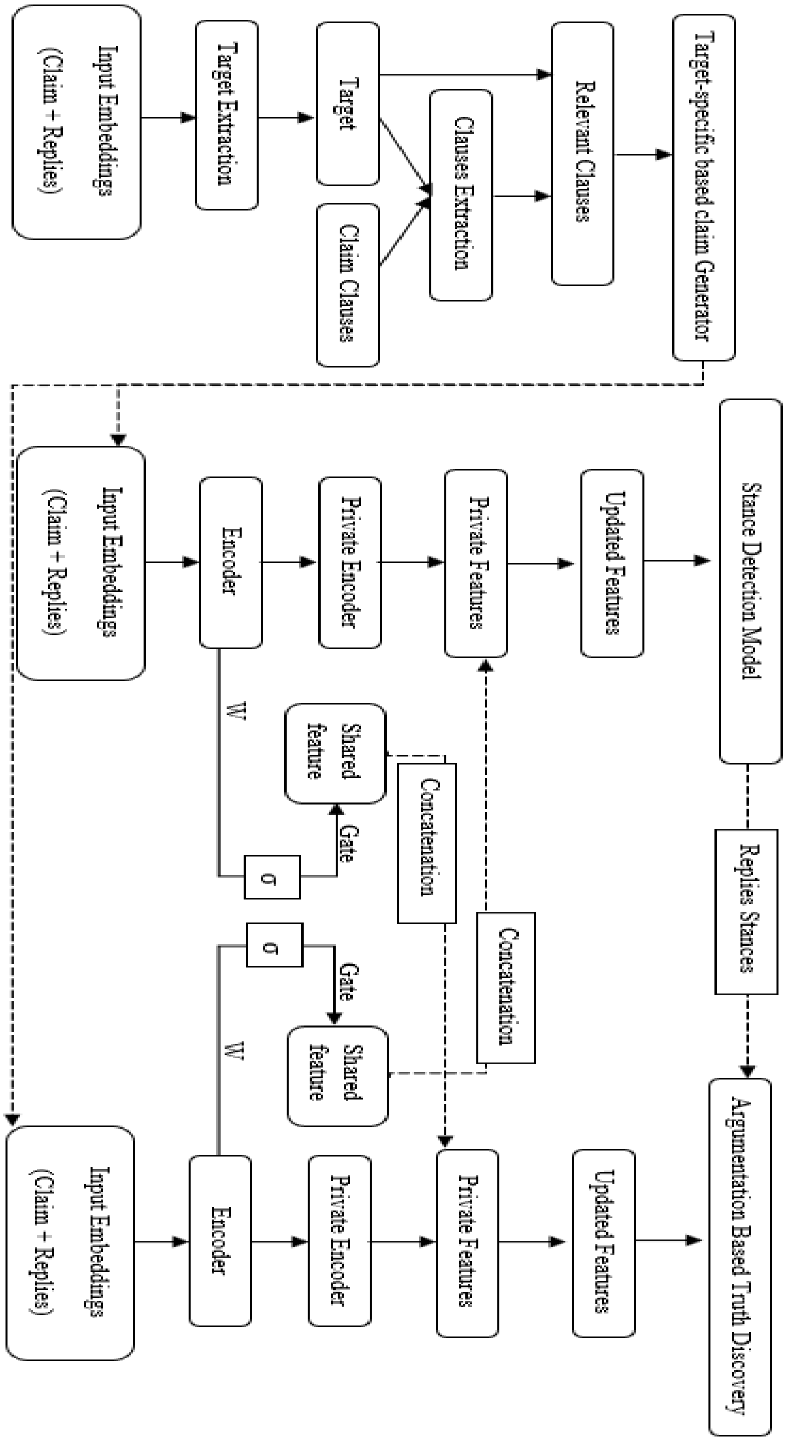


Figure 9.1: The architecture of argumentation-based truth discovery model.

- The target-specific based claim generator component employs a seq2seq architecture with attention and copy mechanisms to generate claims focused on a single target topic.
- The stance detection component was developed to determine the position of the article claim relative to other replies.
- The prediction component is created by using Argumentation-Based Truth Discovery to determine whether the claim is valid.

The input is in the form of a claim accompanied by a subset of tweet replies, each with a distinct stance: for or against. While stance classification entails per-tweet predictions, verification tasks require only a single output for the initial claim.

As in the Emergent dataset, the article's claim source may be longer than the user replies (the related he is supporting and opposing perspectives), and it may also cover a wider range of target topics while the user is only interested in one. For instance, several example candidate target topics from the Emergent dataset article are listed in table 9.1.

The proposed model is intended to extract the user preferred target topics and generate the primary target topic that will be used to generate a more effective target-specific claim.

Following the primary target topic extraction, the most relevant clauses for the claim are extracted, with only informative information on the target topic. The sequence-to-sequence generator receives the selected clauses and directs the generation process toward the target topic. Numerous Evaluators are used to guiding this model's adversarial training using training signals to optimise its parameters, i.e., determining the difference between generated and ground truth target-specific based claims. Finally, before predicting the claim's veracity, the generated target-specific based claims are used to verify the original claim's stance on the response’s claims.

Numerous multi-task neural network models, such as hard parameter sharing networks and soft parameter sharing networks [431]. This paper adopts a soft parameter sharing network model because every task has its network. A gate mechanism will ensure that only beneficial features of auxiliary tasks are shared with the primary task [431]. Filtering feature flows between tasks is accomplished by assigning them a higher weight (learnable parameter) via a gate mechanism that utilises both sigmoid and scalar weights. The gate mechanism produces a vector of elements in the range [0, 1] that can be used to select (or retain only a subset of) the advantageous features required to perform the given task.

Table 9.1 An example of our proposed model-ATD on emergent data [257].

Initial Source	law360.com	
The article	Wonder how long a Quarter Pounder with cheese can last? Two Australians say they bought a few McDonald's burgers for friends back in 1995 when they were teens, and one of the friends never showed up. So, the kid's burger went uneaten and stayed that way, Australia's News Network reports. "We are pretty sure it is the oldest burger in the world," says one of the men, Casey Dean. Holding onto the burger for their friend "started as a joke," he adds, but "the months became years and now, 20 years later, it looks the same as it did the day we bought it, perfectly preserved in its original wrapping." Dean and his burger-buying mate, Eduard Nitz, even took the burger on the Australian TV show The Project last night and "showed off the mould-free specimen" News 9 reports. The pair offered to take a bite of it for charity but were dissuaded by the show's hosts. They have also started a Facebook page for the burger called "Can This 20-Year-Old Burger Get More Likes Than Kanye West?" with more than 4,044 likes as of this writing. Furthermore, they are selling an iTunes song, "Free the Burger," for $1.69, and giving proceeds to the charity Beyond Blue, which helps Australian’s battle anxiety and depression. (A few years ago, a man sold a 20-year-old bottle of McDonald's McJordan sauce for $10,000. Here's why Mickey D's food seemingly, never decays.).”	
Candidate target topics	Australia, Food, Hamburger, McDonald's, Quarter + Pounder …etc	
Extracted primary target topic	McDonald's	
Clause Selection	Wonder how long a Quarter Pounder with cheese can last? Two Australians say they bought a few McDonald's burgers for friends A man sold a 20-year-old bottle of McDonald's McJordan sauce for $10,000.	
Generated target-specific claim	For 20 years, two Australian men held a McDonald's Quarter Pounder with Cheese	
Stance Detection from different sources	**Source-1**: 9 news.com.au **Headline**: Two blokes dared to eat a 20-year-old burger for charity. **Stance**: for **Source-2**: mirror.co.uk **Headline**: Is this the world's oldest burger? Man claims to have kept McDonald's Quarter Pounder for 20 YEARS. **Stance**: for	**Source-3**: examiner.com **Headline**: 20-year-old burger: McDonald's Quarter Pounder looks nearly new after 2 decades. **Stance**: observing **Source-4**: techinsider.net **Headline**: 20-Year-Old Quarter Pounder Looks About the Same. **Stance**: observing
Overall Veracity Prediction via ATD:	true	

Following the embedding layer, a vector is typically used to represent each word in the input. Our model assigns a private sub-model and a private encoder to each task to extract shared and private features from multiple tasks. We begin by calculating the common representation [h1,..., ht] by encoding the tasks' input embeddings with an encoder such as BiGRU. Then, we employ the attention mechanism to selectively retrieve task-specific information and incorporate gates for useful features that transfer between tasks. For each task, both private and shared features are concatenated.

As with encoders, our models incorporate gates to facilitate the transfer of features between sub-models. A gate g is added to task j when it borrows features from task k to select the most useful ones. The gate g is calculated from the previous layer as equation 9.1:

Equation 9. 1 $g_{jk}^{l} = \sigma\left(W_{jk}^{l} \cdot F_{k}^{l} + b_{jk}^{l}\right)$

Where l means the level of the layers and σ denotes the nonlinear activation of the sigmoid. The output F of gates from task j is calculated by fusing the lower layers Fl from all the tasks together, equation 9.2:

Equation 9. 2 $F_{j}^{l+1} = \sum_{k\in C,k\neq j} g_{jk}^{l} \odot F_{k}^{l} + F_{j}^{l}$

We introduce a task-specific query vector q(k) to calculate the attention distribution α(k) overall positions as in equation 9.3.

Equation 9. 3 $a_{t}^{(k)} = softmax\left({q^{(k)}}^{T} h_{t}\right)$

Where the task-specific query vector q(k) is a learned parameter, a task-specific query vector will be used to focus target for conclusion generator and claim for stance detection and rumour veracity. The final task-specific representation c(k) is summarized in equation 9.4.

Equation 9. 4 $c^{(k)} = \sum_{t=1}^{T} a_{t}^{(k)} h_{t}$

9.2.1. Target-specific Based Claim Generator

Target-specific based claim generator is a model that conveys a specific stance toward a specific target topic as a key to understanding an argument from its claim, especially if it is a long one. Figure 9.2 depicts a general overview of this component. Abstractive text summarization is the closest work to this model's target-specific based claim generator; most of them generate summaries by the decoder based on encoded information from the encoder; some use a copy machine to solve the out-of-vocabulary problem [404], [432].

Different earlier approaches [433]–[436] are proposed to capture the central target topic then summarise based on the main target topic. Chen et al. [436] proposed target topic aware summarisation by rewriting the most silent sentences, which achieves the best performance on CNN/Daily Mail benchmark Dataset [437].

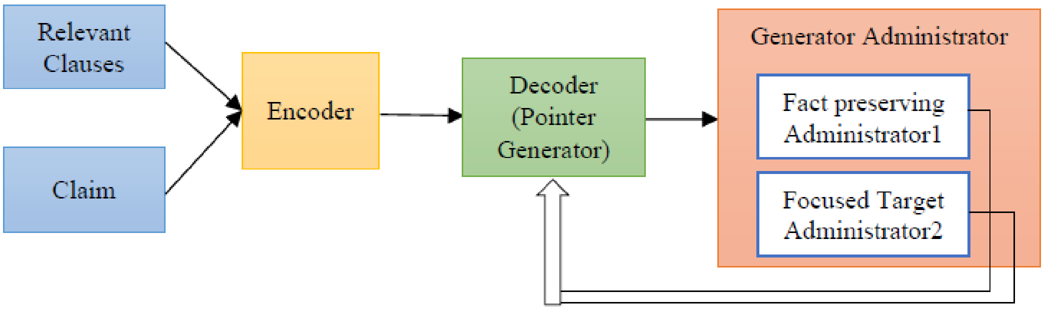

Relevant Clauses
Claim
Encoder
Decoder (Pointer Generator)
Generator Administrator
Fact preserving Administrator1
Focused Target Administrator2

Figure 9.2: Overview of target-specific based claim generator.

This work employs a pointer generator architecture with attention and copy mechanisms to create a claim-target topic-based generator. The pointer generator acts as a decoder, with the selected clauses vectors concatenated with the primary target topic passed to the article encoder serving as the model's inputs. The generator receives the representation outputs from each encoder (decoder). Both encoders and decoders employ a Recurrent Neural Network, namely Bi-GRU encoders and GRU decoders.

9.2.2. Extraction of the Target Topic

Because a strong target-specific based claim should include the primary objective of the claim, we argue that deducing a claimed target topic is a critical step in conducting lengthy based claims and that this targeted topic should be related to the replay's target topics. The extracted target topic is used to generate the target-specific based claim of an argument based on its article, in which all associated replies adopt a single target topic position. The primary target topic is used to demonstrate or indicate the subject to which the author wishes to direct readers, whereas each claim, particularly the longer ones, may cover a variety of topics or convey the same event via a variety of target topics. As a result, a long claim has generated claim should be focused on the primary target topic. Additionally, the target-specific based claim aids in the detection of stances associated with the claim from replies.

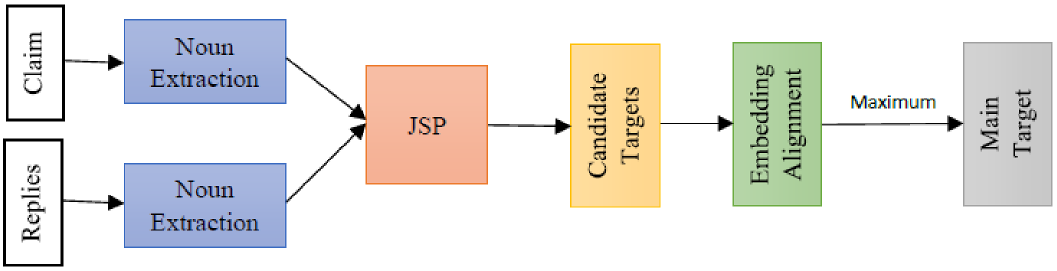


Figure 9.3: The general architecture for claim target topic extraction.

As shown in figure 9.3, the purpose of this component is to extract the primary target topic shared by a claim and its associated replies from among candidate target topics. The nouns and replies nouns in the claim must first be extracted. Each noun must be represented as a vector that includes a probability distribution. The Jensen-Shannon Divergence and a distance score greater than or equal to the threshold, set empirically at 0.75, are then used to identify candidate target topics. The Shannon-Jensen Distance (SJD) is an asymmetric version of the Kullback-Leibler Divergence, which uses the difference measure to compute probability distributions [438] which provides a measure of the distance between two probability distributions [439]. It is used to determine text-similarity [440], such as those represented by the p and q vectors in equation 9.5. The relevance score is calculated based on the distance score:

Equation 9. 5 $1/2(\mathrm{D}(\mathrm{p}\|\mathrm{m}) + \mathrm{D}(\mathrm{q}\|\mathrm{m}))$

Where m = 1/2 (p + q). The two distributions below represent the candidate target topics:
p = as array ([0.10, 0.40, 0.50])
q = as array ([0.80, 0.15, 0.05])
Jensen-Shannon divergence (P || Q): 0.42
Jensen-Shannon distance (P || Q): 0.648, distance is sqrt of divergence.

The JSD [441] explains the contribution of the word I. The smallest divergence indicates that the claims and their replies have a common target topic, equation 9.6:

Equation 9. 6 $D_{JS,i}(P||Q) = -m_i log_2 m_i + \pi_1 p_i log_2 p_i + \pi_2 q_i log_2 q_i$

Where m_i denotes the likelihood that the word I will appear in M. By determining which of p_i or q_i is greater, we can attribute the contribution to the divergence from the word I to text P or Q. The probabilities of seeing the word I in P and Q are p_i and q_i, respectively.

The Jensen-Shannon Divergence was used to select the candidate target'= topics. The maximum alignment score embeddings of nouns are used to record candidate target topics for claim and replies to link the claimed target topic to the argument's replay target topics.

The selected primary target topic has a higher chance of being discussed in the claim and its replies, as explained below. Gao et al. [442] used a max operation over the alignment to select the highly focused noun in the claim by its associated replies, as in equations 9.7 and 9.8. This work determines a claim's semantic word alignment based on its embeddings in the replies to model the claim concentrated target topic. Thus, the alignment score indicates the degree to which the word in a claim is targeted at the replies., where $e(A_i^s)^T$ is word embedding in the claim article, and $e\left(A_{j,n}^c\right)$ is word embedding in the replay, $target_{i,j,n}$ is the attention for the i-th claim word with the j-th replay word, s is a replay, c is claim, n is article number, i is index word of the replay, and j is the index word in the claim.

Equation 9. 7 $\mathrm{TARGET\ TOPIC}_{i,j,n} = \mathrm{e}(\mathrm{A}_i^c)^T\, \mathrm{e}\left(\mathrm{A}_{j,n}^s\right)$

Equation 9. 8 $maximum_{i,j} = max\left(\left\{target_{i,j,1}, \dots, target_{i,j,T_j^c}\right\}\right)$

9.2.3. Clause Selection Model

Clause selection model selects target topic-relevant clauses and eliminates irrelevant and noisy clauses, as our target-specific based claim generator attempts to focus on a single target topic against which other replayed stances can be compared. Clause Picking Module's job is to break down a sentence into clauses and incorporate knowledge of and text information at the clause level. To our knowledge, we are the first to address stance detection by incorporating the critical clause for predicting stances while considering the clauses that correspond to the specific target topic.

Following the first module's selection, the clause selection module selects several clauses about the main target topic, discarding irrelevant and noisy clauses mentioning other target topics, and then generates the claim. When noisy clauses for other target topics are provided and taken into account, the model may make mistakes. The goal of this component is to retrieve the most target topic-relevant clauses while ignoring the rest.

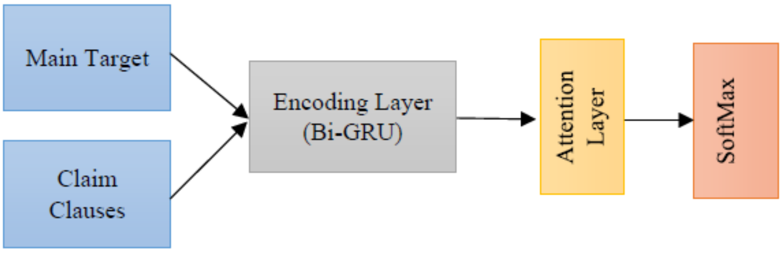


Figure 9.4: The architecture of clause selection model

This component is made up of two layers. The encoding and attention layers are depicted in figure 9.4, which employs bi-directional GRU to capture context clause representations for each clause relevant to the main target topic by concatenating the target topic and each clause cl in the claim. To learn the hidden semantics of words, this model employs GRU, which is more efficient than LSTM training [440]. This module employs two GRU neural networks: a forward GRU and a backward GRU, which process the sentence from left to right and reverse order, respectively, and handle the word vectors in order. Finally, the forward-GRU and backward-GRU units are concatenated to learn the claim's bidirectional semantics and each clause in the article to emphasise the claim's importance. Then, the attention mechanism is used to capture the valuable information the article clauses. The encoding layer does this for both clauses and target topic (h_i for target topic h_j for clauses).

The attention layer produces clause vectors by focusing on words that are relevant to the target topic context. Attention methods would focus on the terms in the clause that are concatenated with the target topic and combine their representations to form a clause vector. The clause representation of the target topic is indicated by clr. The attention mechanism decides the weight assigned to each target topic-clause representation to produce contextual information based on the target topic representation. The weight of each clause concerning its representation of a specific target topic will be determined using equations 9.9-9.12:

Equation 9. 9 $a_{vi} = avg\left(\left\{h_{i,1}^{c}, h_{i,2}^{c}, \ldots, h_{i,T_i^c}^{c}\right\}\right)$

Equation 9. 10 $m_i = tanh\left(W_{cl} \cdot \left[a_{vi}; h_{j,T_j^{cl}}^{cl}\right] + b_{cl}\right)$

Equation 9. 11 $a_i = softmax(m_i) = \frac{exp(m_i)}{\sum_{t=1}^{Cl} exp(m_t)}$

Equation 9. 12 $clr = \sum_{i=1}^{|cl|} a_i \,.\, h_j^{cl} \cdot$

$h_{j,T_j^c}^{cl} l$ is the state of the clause, cl is clause and c is a target topic, a_{vi} is the average of hidden states for the target topic, a_i is attention weights, and the clause representation clr is calculated based on the attention vectors a_i. In this model, to select relevant clauses for the target topic, conditional probability using SoftMax Layer is used to perform the target topic clause's relevant classification. Then, feeding the clause representation clr to a SoftMax classifier. This model is trained by cross-entropy, W and b are the parameters for the model. W is the weight matrix, and b is the bias. The final output o is obtained by equation 9.13.

Equation 9. 13 $o = W^{*}\, clr + b$

Loss function computed by cosine similarity between target topic embedding and hidden state of the t-th clause. The similarity is the relatedness between each word annotation hij and the Target' topic representation.

9.2.3.1. Article (Relevant Clauses) and Claim Encoder

A bidirectional GRU is used to get both the context before and after the word. first, the Word Embeddings method implemented in this work is Glove [146] and word2vec [140], which work better. The encoder generates a state for each input word by BiGRU for the target topic and claims to obtain the context representation around a word. T̃he claim includes all relative clauses retrieved by the clause selection model. $GRU_{\overrightarrow{evi}}\ and\ GRU_{evi^{\leftarrow}}$ are the forward and backwards representation, h_i denotes hidden state. $[\vec{h}_i, h^{\leftarrow}{}_i]$ is the merge of the forward and backward hidden state, *evi* is the claim and c is the target topic. h_i and h_j are the annotations for claim and target topic. It used to compute a hidden representation for claim from both directions and the same for claim encoding.

9.2.3.2. Decoder

The decoder employs unidirectional GRU, with each decoder time step receiving claim concatenated with its Target' topic representation as input. The decoder begins the decoding process to generate the target-specific based claim from the claim based on the input encoder's final state and target topic representation. At each decoder time step, the target topic embedding is fed as input to allow the decoder to change the output sequence and generate a statement about the primary Target' topic.

The word distribution is calculated, and the word with the highest probability on the decoder state and context vector output is chosen using the SoftMax function. At each decode time step, a sigmoid activation function is used to choose between two options: copy from the original input or generate from the vocabulary, so is the final article encoder state, Ct is the context vector at time step t from the attention mechanism, and Yt-1 is the predicted output word at time step -1. The attention mechanism identifies the input's relevant parts by learning the decoder to focus on different portions of the claim and target topic at different time steps [413]. This could be accomplished using equations 8.14-8.16, inspired by Nema et al. [443], modified to conform to the proposed model. The attention mechanism for the evidence is applied to help the decoder output focused claim tokens at each step using equations 9.17 and 9.18 whereα_{ti} represents weights to each in the claim at each decoder timestep, St is the current state of the decoder at time step t. The final claim representation at time step t is computed in equation 9.19. Attention mechanism for the claim which assigns weights to each word in the claim at each decoder timestep, h^c_j is claim word hidden states. The final claim representation at time step *t*, which is computed as:

Equation 9. 14 $a^c_{t,j} = v_{cl} \cdot tanh\left(W_{cl}s_t + U_{cl}h^c_j\right)$

Equation 9. 15 $\alpha^c_{tj} = \frac{exp\left(a^c_{t,j}\right)}{\sum_{j=1}^{|cl|} exp\left(a^c_{t,j}\right)}$

Equation 9. 16 $c_t = \sum_{i=1}^{|c|} \alpha^c_{t,i}\, h^c_j$

h_i hidden representation for each time-step for evidence word I, $E_{(Y_{t-1})}$ is the previous word embeddings, st is the current state of the decoder at time step t, and cct is the final claim representation at a time step. The hidden state of the decoder st at each time t is computed as follow, considering the previous state s_{t-1}, the embedding distribution of the claimed Target' topic $target$, the previous claim context vector c_{t-1} and the previous word embeddings, the state is defined as equation 9.17:

Equation 9. 17 $s_t = GRU_{dec}\left(s_{t-1}, \left[c_{t-1}, target, E_{(Y_{t-1})}\right]\right)$

The probability distribution over the output vocabulary o_t, as equation 9.18 to decide the word which has the highest probability is computed from the context vector ct, the decoder state st as:

Equation 9. 18 $o_t = W_g^{(2)}\left(W_g^{(1)}[s_t, c_t] + b_g^{(1)}\right) + b_g^{(2)}$

Inspired by Hasselqvist et al. [423], The decoder in this work uses the pointer mechanism to decide whether to copy from the original document or generate the vocabulary based on the pointer output; the next word can then be chosen. $p_t^{pointer}$ is used as a switch to select between (a) copying words from the source text via pointing (copying a word from the input sequence by selection according to the attention distribution) or (b) generating a word from the vocabulary by selecting based on Pv in equation 9.19.

Equation 9. 19 $p_t^{pointer} = sigmoid\left(v_{ptr}^T\left[s_t, E_{(Y_{t-1})}, c_t\right] + b_{pointer}\right)$

The generation probabilityp_{tj}^{gen}∈ [0,1] for timestep t is computed as equation 9.20. If p_{tj}^{gen} > 0.5, word is copied from the input determined by the attention distribution where the attention is the highest, else the generator output is used. The probability of generating timestamp t is set to 0. 5 empirically.

Equation 9. 20 $p_{tj}^{gen} = \frac{exp(o_{tj})}{\sum_k exp(o_{tk})}$

The model then generates distribution Pv over vocabulary. It concatenates on Evaluators 1, and 2, (details below) and the decoder's output to guide the decoder. $P_v\ is$ probability distribution over all words in the vocabulary and gives us the final distribution to expect words. It concatenates Evaluators $evalt_t; evalt2_{t;}$ and the output of decoder st as the input of the output projection layer. The goal of, $evalt_t$and$evalt2_{t;}$is to keep track of the difference between the generated target-specific based claim and the focused Target' topic and the fact that in the next subsection, it will show the details of these variables in equation 9.21.

Equation 9. 21 $P_v = softmax\left(W_v\left[s_t; E_{(Y_{t-1})}, c_t; evalt_t; evalt2_{t;}\right] + b_v\right)$

This model uses ROUGE scores to evaluate the generated target-specific based claim's quality [415].

9.2.3.3. Generators' Evaluators

This work employs two evaluators allowing the discriminator to provide additional information, denoted as Evaluators 1 and 2, to reduce the noisy and irrelevant generated words by the decoder and guide the generator to focus on the information related to the target topic claim and preserve the fact information. Thus, two Evaluator modules guide the generator through the decoding process, preserving the fact while focusing on the claim's primary target topic. Another advantage of target topic extraction is that it can determine which rumour should be believed among several conflicting claims, rather than comparing the veracity of rumours to their replay stances individually, as other experiments do. The discriminator is a binary classifier that uses a convolutional feature extractor and a sigmoid classification layer to signal the generator. The two Evaluators' details are explained in turn below.

Evaluator-1 to Check Decoder Focused Target Topic: It is a discriminator that extracts features from a convolutional neural network and then compares the decoder-focused target topic (target-specific based claim) to the claim-focused target topic. It quantifies the semantic difference between the decoder-focused target topic and the claimed target topic; for example, the claimed target topic is "McDonald's Quarter," but the generated target-specific based claim may emphasise the "charity" target topic.

This model makes use of element-wise difference to simulate the difference between target topic and target-specific based claim attention and then uses the decoder to identify the unfocused target topic. The decoder then uses the difference between the attention distribution and the weighted sum of the document states as the context vector to assist it in producing a target-specific based claim that is more focused on the target topic.

To persuade the generator to focus on the claimed target topic, this model employs a CNN-based discriminator to represent the difference between the generator- and claim-focused target topics. After concatenating the target topic, the sentence vector is generated using the BiGRU word-level encoding module. The model establishes a Bi-directional GRU (Bi-GRU) network taking the sentence representation as input further to study the interactions and information exchanges between sentences. This architecture allows information to flow back and forth to generate new sentence representations. The attention-based CNN model for this evaluator will be used. The target-specific based claim's final target topic representation is fed into an output layer to predict the probability distribution on the target topic, which is defined as Adm1 via equations 9.22-9.30. It is trained via cross-entropy minimisation for training target topic-based target-specific based claim generation.

Equation 9. 22 $target = \frac{1}{m}\sum_{i=1}^{m} e_{x_i}$

The attention vector decides which semantic features in each hidden state are meaningful specifically towards the Target' topic, which is calculated through a gated structure, as follows:

The focused Target' topic for the original claim

Equation 9. 23 $\mathrm{score_i} = \tanh\left(\mathrm{h_i^T W_1 target\ topic}\right)$

Equation 9. 24 $\mathrm{\alpha ttention_i} = \frac{\exp(\mathrm{score_i})}{\sum_{j=1}^{n}\exp(\mathrm{score_j})}$

We sum up the all attention distributions {αt , αt−1 , . . . , αt−n+1 } and result in vt1

Equation 9. 25 $\mathrm{vt1} = 1/\mathrm{n}\sum_{i=1}^{n}\mathrm{\alpha ttention_i}$

Equation 9. 26 $\mathrm{St1} = \sum_{i=1}^{n}\mathrm{vt1_{,i}\ h_i}$

$St1$ represents the focused Target' topic for the original claim

The focused Target' topic for the generated target-specific based claim.

Equation 9. 27 $score_i = tanh\left(h_i^T W_1 st\right)$

Equation 9. 28 $\alpha ttention_i = \frac{exp(score_i)}{\sum_{j=1}^{n} exp(score_j)}$

We sum up the latest k attention distributions {α_t , α_{t-1} , . . . , α_{t-k+1} } and result in vt2

Equation 9. 29 $vt2 = \sum_{i=1}^{k} \alpha ttention_i$

Equation 9. 30 $St2 = \sum_{i=1}^{n} vt_{,i}\, h_i$

$St2$ represents the focused target topic by the latest k decoding steps, a.k.a., decoder focused target topic.

The score is a content-based function that encapsulates the semantic relationship between the decoder output target topic and the claimed target topic used to determine each word's relative importance in the target-specific based claim and claim. St is used to encode the relevant information n from the target-specific based claim and the claim. The target topic is represented by the target topic's averaged word embedding.

In order to model the gap between claim-focused target topic and decoder focused target topic, we subtract the claim attention by decoder attention resulting in the attention difference shown in equation 9.31. Then, we use the attention difference to sum up the document hidden states h, 9.32.

Equation 9. 31 $diff = vt1 - vt2$

Equation 9. 32 $evalt_t = \sum_{i=1}^{n} diff_{,i}\, h_i$

Evaluator 2 to Check the Fact Preserving: This model applies a denoising autoencoder to evaluate that preservation is related to the target-specific based claim's fact concerning the claim' source, i.e., integrating knowledge from the source article; this is represented as a factual score. After extracting the fact related to the source article's target topic, it applies BiGRU to extract hidden states for both facts in the article and the generated fact. At each decoding time step t, GRU reads the previous output yt−1 and context vector ct−1 as inputs to compute the new hidden state st. Then the context vectors are computed. The fact vector equations 9.33-9.35 are applied, and equations 9.36-9.38 for a generated fact. Besides the decoder's current state, a combination of both context vectors is used to guide the decoder to generate more factual words.

Attention mechanism for the facts in the original claim

Equation 9. 33 $e_{t,i}^{fact} = MLP\left(\mathrm{s_t}, \mathrm{h_i^{fact}}\right)$

Equation 9. 34 $\mathrm{a_{t,i}^{fact}} = \frac{\exp\left(\mathrm{e_{t,i}^{fact}}\right)}{\sum_{\mathrm{j}} \exp\left(\mathrm{e_{t,j}^{fact}}\right)}$

Equation 9. 35 $\mathrm{c_t^{fact}} = \sum_{\mathrm{i}} \mathrm{a_{t,i}^{fact}}\, \mathrm{h_i^{fact}}$

Attention mechanism for the generated target-specific based claim

Equation 9. 36 $e_{t,i}^{gen} = \mathrm{MLP}\left(s_t, h_i^{gen}\right)$

Equation 9. 37 $a_{t,i}^{gen} = \frac{\exp\left(e_{t,i}^{gen}\right)}{\sum_j \exp\left(e_{t,j}^{gen}\right)}$

Equation 9. 38 $c_t^{gen} = \sum_i a_{t,i}^{gen}\, h_i^{gen}$

Context vectors are merged by using equations 9.39-9.42.

Equation 9. 39 $c_t = \left[c_t^{fact}; c_t^{gen}\right]$

Equation 9. 40 $s_t = \mathrm{GRU}(Y_{t-1}, c_t, s_{t-1})$

Equation 9. 41 $m_i = \tanh\left(W_1 \cdot \left[[s_t, c_t]\right] + b_1\right)$

Equation 9. 42 $\mathrm{output}_i = \mathrm{softmax}(m_i) = \frac{\exp(m_i)}{\sum_{t=1}^{C} \exp(m_t)}$

The unfocused hidden state of the decoder is detected in equations 9.43 and 9.44.

Equation 9. 43 $f_t = c_t^{fact} - c_t^{gen}$

Equation 9. 44 $\mathrm{evalt2}_{t;} = \sum_{i=1}^{T^d} f_t\, , h_i^d$

The scalar training signals from the discriminator is based on equations 9.45 and 9.46.

Equation 9. 45 $\mathrm{Adm1} = \mathrm{sigmoid}\ (\mathrm{W.\ evalt1} + \mathrm{b})$

Equation 9. 46 $\mathrm{Adm2} = \mathrm{sigmoid}\ (\mathrm{W.\ evalt2}_2 + \mathrm{b})$

Adm1 is the scalar training signal for generator training, which means there is still unfocused generated information; adm1 is fed to target-specific based claim generator helping to reduces the unimportant information and focus more on the main target topic. Adm1 maximises the probability of generating a target-specific based claim toward a target topic. W is the weight matrix, and b is the bias. Adm2 is a scalar training signal for generator training; it ensures that the original article's fact is preserved. Adm2 is fed to the target-specific based claim generator, which helps to avoid changing the fact. Adm2 represents the gap content between the factual and non-factual generated target-specific based claim. The gap guides the generator to preserve the fact. The probability of generating the next word is based on the SoftMax layer result.

9.2.4. Stance Detection

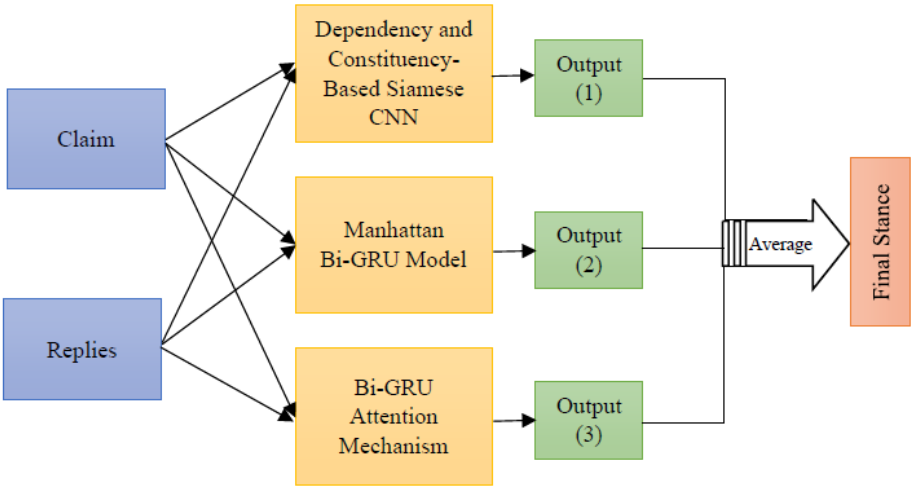


Figure 9.5: The proposed stance detection model.

The current model uses three methods to detect the generated target-specific based claim's stance toward all replies, as shown in figure 9.5. Method 1 uses a dependency and constituency-based Siamese CNN to detect stance, Method 2 uses Manhattan distance to detect stance, and Method 3 uses an attention mechanism to detect stance. All possible outcomes of models are considered to arrive at a final prediction. The calculation could be performed with each model contributing an equal amount to the ensemble prediction or with each model contributing a different outcome based on its contribution to the ensemble prediction's weighting. The method of weighted ensemble outperforms the method of the equal-weighted ensemble. Ensemble techniques are used to detect stances, which aggregate our baseline classifiers. Ensemble methods are based on the concept of combining multiple models (base classifiers) to create a more accurate and reliable model than a single model can provide. As a result, each model is weighted differently.

9.2.4.1. Method 1 (for Stance Detection): Dependency and Constituency-based Siamese CNN

This work extracts sentence-level features from the generated target-specific based claim and replies. It suggests the Stanford Parser be used to perform constituency and dependency parse on the inputs to extract the most important sentence information, such as the main linguistic structure that provides information, for example, the subject, predicate, and object of a sentence that are the essential roles in a sentence. It is necessary to learn better sentence representations to observe the structure of sentences and the relationship between the words for each of them.

Word sequences based on their text's constituency words are concatenated with their dependency parser-based word sequences as an input to this work's CNN-based model, where each of them is fed to convolution operations separately for both claim and target-specific based claim. The convolution operation applies a filter w and bias as in equation 9.48 with sigmoid function to words representing constituency or dependency that occur in the sentence, and they are all concatenated to represent the entire feature map for all the words in the sentence. This model combines information for both constituency and dependency-based CNN models to exploit more vital information and capture different features. Concatenate these representations to produce the statement (claim or target-specific based claim) representation, the sum of all vectors. A convolutional neural network is used to transform the generated target-specific based claim representation from word embedding vectors to semantic sentence hidden states. Then, to reduce the representation's spatial size while retaining essential features, a pooling operation is used. The attention vector is generated by connecting the target topic and target-specific based claim representation feature vectors into one vector. The matching distance is used to determine how similar target-specific based claim a and a claim replies [410].

Constituency based CNN for replay $\boldsymbol{X_c}$: For a given claim, the convolution operation applies a filter W for each constituency based on concatenated word sequence X_c (or words with is constituent structures words from the constituency sub-tree) from the claim where b_c is the bias as equation 9.47.

Equation 9. 47 $\mathrm{Y_{ci}} = \mathrm{sigmoid}(\mathrm{W_c X_c} + \mathrm{b_c})$

All words constituency information is concatenated to generate the feature map const, as in equation 9.48:

Equation 9. 48 $\mathrm{const}= \mathrm{Y_{c1}}\,, \mathrm{Y_{c2,}}\mathrm{Y_{c3}}\ \mathrm{Y_{cl}}$

Dependency-based CNN for replay $\boldsymbol{X_{dep}}$: For a given claim, the convolution operation applies a filter W for each dependency-based concatenated word sequence X_{dep} (or words with is dependent structures words from the dependency sub-tree) from the claim where b_c is the bias as equation 9.49:

Equation 9. 49 $\mathrm{Y_{depi}} = \mathrm{sigmoid}(\mathrm{W_c X_{dep}} + \mathrm{b_c})$

All word dependency information is concatenated to generate the feature map dep, equation 9.50:

Equation 9. 50 $\mathrm{dep} = \mathrm{Y_{dep1}}\,, \mathrm{Y_{dep2}}\mathrm{Y_{dep3}}\ \mathrm{Y_{depl}}$

Dependency-based CNN and constituency-based CNN concatenation for replay: The max-pooling operation is applied for both feature maps, const. and dep., to extract the most significant features from each of them, then they are concatenated as equation 9.51:

Equation 9. 51 $\mathrm{max1} = \mathrm{max}\ (\mathrm{dep}) + \mathrm{max}(\mathrm{const})$

The same equations, equations 9.47-9.51, used for claim feature representation, will be used for the target-specific based claim, equations 9.52-9.56.

Constituency-based CNN for the target-specific based claim X_e:

Equation 9. 52 $\mathrm{Y_{ei}} = \mathrm{sigmoid}(\mathrm{W_e X_e} + \mathrm{b_{e)}}$

Equation 9. 53 $\mathrm{e}= \mathrm{Y_{e1}}\, , \mathrm{Y_{e2}} \mathrm{Y_{e3}}\ \mathrm{Y_{en}}$

Dependency-based CNN for the target-specific based claim X_{dep2}:

Equation 9. 54 $\mathrm{Y_{depi}} = \mathrm{sigmoid}(\mathrm{W_e X_{depi}} + \mathrm{b_{ce}})$

Equation 9. 55 $\mathrm{dep2} = \mathrm{Y_{dep1}}\, , \mathrm{Y_{dep2}} \mathrm{Y_{dep3}}\ \mathrm{Y_{depl}}$

Dependency-based CNN and constituency-based CNN concatenation for the target-specific based claim:

Equation 9. 56 $\mathrm{max2} = \mathrm{ma\,x}(\mathrm{dep2}) + \mathrm{max}(\mathrm{const})$

After generating the vector representation of sentences, three matching methods are applied to extract relations between ($max1$; $max2$)

1. Concatenation of individual representation ($max1$; $max2$) to produce r1
2. Element-wise product ($max1$* $max2$) to produce r2
3. Absolute element-wise difference ($max1$- $max2$) to produce r3

All the resulting vectors r1, r2, and r3 are concatenated and fed to a SoftMax classifier to predicts the stance label between Claim and target-specific based claim as equation 9.57:

Equation 9. 57 $\mathrm{f} = \mathrm{r1} \oplus \mathrm{r2} \oplus \mathrm{r3}$

f is a fully connected neural network, equation 9.58.

Equation 9. 58 $\mathrm{output1} = \mathrm{softmax}(\mathrm{W_1 f} + \mathrm{b_1})$

9.2.4.2. Method 2 (for Stance Detection): Manhattan- Bi-GRU Model

Bi-GRU is applied to extract the final hidden state's representation, a vector representation for each claim (replay) and target-specific based claim and then use them to compute the semantic similarity between them. The semantic similarity is computed by the Manhattan Bi-GRU Model [410], as equation 9.59, where the distance is transformed into a similarity score to measure the strength of the claim toward the target-specific based claim. $h^{(c)}$ and $h^{(e)}$ are the last hidden representations for the claim and target-specific based claim respectively.

Equation 9. 59 $output\ 2 = exp\left(-\|h^{(c)} - h^{(e)}\|_1\right)$

9.2.4.3. Method 3 (for Stance Detection): Bi-GRU Attention Mechanism

Attention mechanisms capture the most relevant features to detect the target-specific based claim's stance toward the primary target topic. This model merges them as one vector after extracting the hidden states for Bi-GRU's target topic and target-specific based claim. The word attention weights are computed using equations 9.60-9.64, where $h^i_{conc}\ and\ h^i_{claim}$ are the average of hidden states from Bi-GRU for the target-specific based claim and claim, respectively, $\alpha_i\ and\ \beta_i$ are attention vectors for both claim and target-specific based claim that are used to compute word attention weights. Then the text representation considers the common features between them as in equation 9.65:

Equation 9. 60 $\mathrm{att}\left(\mathrm{h}^{\mathrm{i}}_{\mathrm{conc}}\right) = \tanh\left(\mathrm{h}^{\mathrm{i}}_{\mathrm{conc}} \cdot \mathrm{W}_1 + \mathrm{b}_1\right)$

Equation 9. 61 $\alpha_{\mathrm{i}} = \frac{\exp\left(\mathrm{att}\left(\mathrm{h}^{\mathrm{i}}_{\mathrm{conc}}\right)\right)}{\sum_{\mathrm{j}=1}^{\mathrm{n}+1} \exp\left(\mathrm{att}\left(\mathrm{h}^{\mathrm{i}}_{\mathrm{conc}}\right)\right)}$

The following equation 9.62 is applied to generate the final representation for the target-specific based claim representation with Target’s topic.

Equation 9. 62 $\mathrm{target - specific\ based\ claim}_{\ \mathrm{r}} = \sum_{\mathrm{i}=1}^{\mathrm{n}+1} \alpha_{\mathrm{i}}\, \mathrm{h}^{\mathrm{i}}_{\mathrm{conc}}$

For the claim concatenated with the target topic, the attention vector is calculated by equations 9.63-9.66:

Equation 9. 63 $att\left(h^i_{claim}, h^p_{conc}\right) = tanh\left(h^i_{claim} \cdot h^p_{conc} \cdot W_2 + b_2\right)$

Equation 9. 64 $\beta_{\mathrm{i}} = \frac{\exp\left(\mathrm{att}\left(\mathrm{h}^{\mathrm{i}}_{\mathrm{claim}}, \mathrm{h}^{\mathrm{p}}_{\mathrm{conc}}\right)\right)}{\sum_{\mathrm{j}=1}^{\mathrm{m}} \exp\left(\mathrm{att}\left(\mathrm{h}^{\mathrm{i}}_{\mathrm{claim}}, \mathrm{h}^{\mathrm{p}}_{\mathrm{conc}}\right)\right)}$

Equation 9. 65 $\mathrm{claim}_r = \sum_{i=1}^{m} \beta_i h_{\mathrm{claim}}^{i}.$

Equation 9. 66 $\mathrm{output3} = \mathrm{softmax}([\mathrm{claim}_r \oplus \mathrm{target} - \mathrm{specific\ based\ claim}\ _r]\ )$

The final stance fs of each article is based on the average of output1, output2 and output3. Where w1+w2+w3=1, weights contribute equally 33.33% weight for each of the models Fs=w1* output1+ w2* output2+ w3* output3. The weighted average ensemble is used to improve prediction scores. Fs=w1* output1+ w2* output2+ w3* output3, w1=0.6, w2=0.2, w3=0.2 that have been empirically established and have a higher predictive performance on final prediction

9.2.5. Argumentation-Based Truth Discovery

Zhao et al. [255] presented preliminary steps towards truth discovery methods based on bipolar argumentation, where a truth discovery network is mapped to a bipolar argumentation framework by assigning a trust score to each source and a belief score to each claim. Cayrol & Lagasquie-Schiex [22] also suggest linking Truth Discovery with Bipolar Abstract Argumentation. They consider a truth discovery network as disjoint sets S, O and F, representing sources, target topics and claims, respectively. For argumentation frameworks, they consider that arguments interact through attacks and support relations. They provide an example as in figure 9.6 to illustrate graph representation of a truth discovery network where sources are s, t, u, v, target topics are o, p and claims are f, g, h, i. For the claims related to target topics o and p, the sources s and t have contradiction views toward f, h while the sources u and v agree source s particularly t on target topic p. They propose a truth discovery operator that assigns each source a trust score and each claim a belief score. Argumentation-based Truth Discovery is inspired by abstract argumentation by identifying such arguments with the sources and claims. They propose to encode source trustworthiness by introducing an argument, e.g., “s is a trustworthy source", and introduce an argument “f is a believable claim" for identifying claim believability. According to the example, B(N) yields two meta-arguments where each argument attacks the other: X1, X2, where X1 = {s, f, h}, X2={t, u, v, g, i}

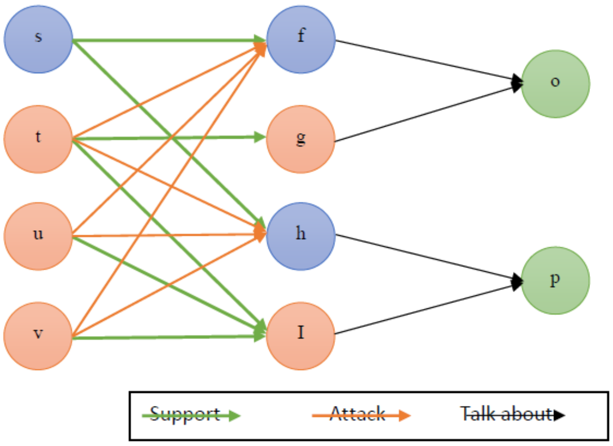


Figure 9.6: Graph representation of a truth discovery network [323], s and t disagree on the fact f, h for target topics o and p. Sources u and v do not comment on claim g but agree with t on target topic p

The proposed work applies an argumentation-based truth discovery, as figure 9.7 and 9.8, where different arguments support contrary target-specific based claim s for specific information from multiple sources with different degrees of trustworthiness, e.g., f and g are contrary from conflict sources, s and t on a particular target topic p.

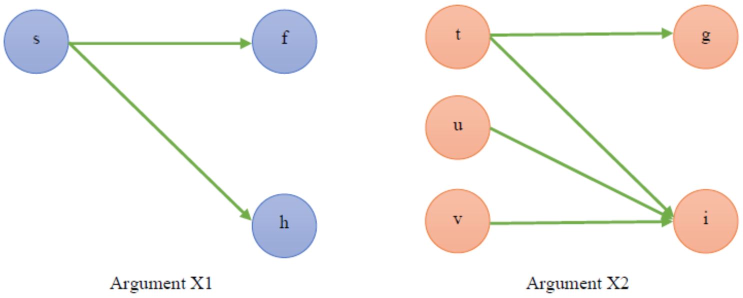


Figure 9.7: Argumentation-based truth discovery: meta-arguments X1, X2

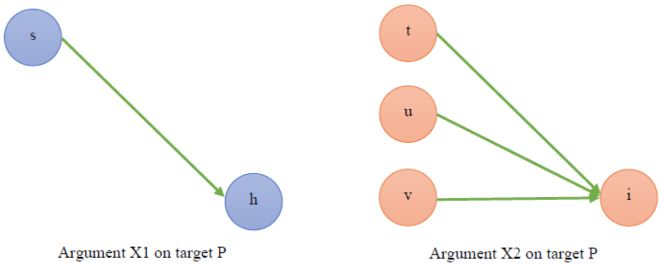


Figure 9.8: Argumentation-based truth discovery: arguments X1, X2 on target P

For two meta-arguments X1, X2, where each argument attacks the other, including the sources with their supported claims, estimation source reliability weight is applied to compute the strength of supporting compared to attacking. First, each meta-argument, including sources with its supported claim, is expressed, e.g., X1={s,f,h} , X2={ t,u,v,g,i. For each target topic, e.g., on target topic p, the candidate truth claims and reliable sources are put in a set for both attacking arguments, e.g., Arg1={s, h}, Arg2={t, u, v, i}. As a result, the strength of the arguments is calculated as follows to select a target topic's truth claim.

1. *Relevance score:* this score measure how the claim and its supporting replies, are relevant and cover the same issue. For each pair of claim and supporting replies, word embeddings in the same meta-argument, the Siamese adaptation of the Long Short-Term Memory (LSTM): Manhattan LSTM model [410] is used because….This model employs an LSTM, with each claim and source represented by the final hidden state. The semantic similarity between them is then determined. After that, all of this model's outputs are averaged for all claim-source pairs.
2. *The dependency between the data sources scores*: The data sources are interdependent, and there is no conflicting information or inconsistency: conflicting claims frequently prevent people from reaching the same target-specific based claim based on the same evidence. The highest correlated source with other sources is computed in the same way as the claim-source pair. The difference between all sources and its supported sources vector and other vectors from other sources, such as the Manhattan distance, is averaged, compared, and ranked (reliable to unreliable). This model computes the probability of correlating with other sources in each meta-argument; u and v are vectors of different sources, equation 9.67.

   Equation 9. 67 $p(s_i|v_i, u_i) = \frac{exp(v_i,u_i)}{\sum_\mathrm{i}\ exp(v_i,u_i)}$

   If the probability is >=0.5, then the source is selected as a candidate trustworthy source. The sources with more correlation and dependency with other sources are considered as a trustworthy source
3. *Interpretation score:* replay should justify the claim to interpret its acceptability. Greater weight is placed on the reasons for accepting a target-specific based claim: the more likely it is that the target-specific based claim is true. This model calculates the probability of each claim in each meta-argument being supported by its sources, equation 9.68:

   Equation 9. 68 $\mathrm{p(s_i|c_i, u_i)} = \frac{\mathrm{exp(c_i,u_i)}}{\sum_\mathrm{i} \mathrm{exp(c_i,u_i)}}$

$p(s_i)$ is the probability of a source u supporting a claim c, i.e., to what extent its associated sources support the claim, u_i is the source vector representation and c_i is the claim vector representation. If the probability is >= 0.5, then the claim is selected as a candidate truth.

4. Sufficient sources: To accept the claim based on a specific target, there must be a sufficient number of relevant and acceptable premises.
5. Conflict sources: That is, this is an argument for controversial issues. A strong argument includes an effective rebuttal to the argument. The provided argument addresses the strongest counterarguments effectively.
6. *Consistency:* Replay embeddings are averaged as a ground truth claim for each meta-argument. The claim with the highest similarity to the average is considered more likely as a truth claim from this argument with its supporting sources.
7. *Argument style:* Finally, as shown in Barrón-Cedeño et al. [115], sufficient vocabulary richness and readability features are used to determine both arguments' most trustworthy source and truth claim. All feature results are weighted to determine the final veracity label. According to Potthast et al. [36], hyperpartisan outlets have a different writing style than mainstream news outlets. Rashkin et al. [1] investigated the relationship between words from the lexicons above in various news articles. They discovered that certain words from their lexicons (swear, see, and negation) appear more frequently in propagandistic, satirical, and hoax articles than in reliable news articles.

Let the index for each argument about a specific target topic be I = 1,..., n. The candidates assert that they have an equal chance of getting it right. For each argument representation, the source and replies are represented by two vectors, content embeddings and processed user profile information embeddings, as Liu et al. (Liu et al., 2015) described. All of the inputs are merged with all of the features described above as input representations. Shared parameters are learned to classify each argument independently, yielding the logits: Ri= θ [input representations], Ri … Rn are then concatenated and passed through SoftMax to determine a probability distribution over all arguments.

Given the prediction of all tasks, a global loss function forces the model to minimise the cross-entropy of the prediction and true distributions for all tasks, equations 9.69 and 9.70.

Equation 9. 69 $\mathcal{L} = \sum_{i=1}^{N} \lambda_i \, L(\hat{y}_i, y_i)$

Equation 9. 70 $L(\hat{y}_i, y_i) = y_i log\hat{y}_i + \ (1 - y_i) log(1 - \hat{y}_i)$

Where λ_i is the weight for task i, and N is the number of tasks, stance detection and veracity prediction.

9.3 Experiments and Results

The baseline approaches are included to facilitate a thorough comparison of our approach. The experiments aim to answer the following question, RQ-6 To what extent can the bipolar argumentation framework be a potential solution for multitask truth discovery problems and improve false information detection performance for conflict claim. We examine if our proposed model ATD outperform the baseline models. Also,what is the effect of each module in ATD on performance improvement. For example, to what extent could generating the target-specific based claim of an argument for a specific target topic aid in inferring stances from related replies? If you select the informative clauses that correspond to a specific target topic and ignore the noisy clauses, you will come up with a better target-specific based claim that conveys a pro or con stance on replies. This model supposes that it is good to focus on the relevant parts of an article for stance detection and then predict the overall veracity. How important is it for a fact-preserving evaluator and a focused evaluator to produce a better summary of the main target topic? To which extent ensemble learning helps to improve the stance detection results? And finally, to what extent are argumentation theories and comparing the strength of arguments beneficial in predicting the truth?

9.3.1. Datasets

For contradictory claims, the SemEval-2019 Task 7 dataset was developed by Gorrell et al. [286]. Finally, the Emergent dataset was developed by Ferreira and Vlachos [257] with 300 claims 2,595 associated news articles.

Macro-averaged F1 has been used as the evaluation metric for the two tasks in RumourEval 2019 [286], as discussed in section 2 above. The RumourEval dataset includes 325 rumorous conversation threads with a training/development/testing split. Additional experiments are conducted on Emergent publicly available datasets [257] as discussed in section 2 above since news article headlines, and evidence of news article content are essential information for this model training. Since the Emergent dataset is the largest, collected from a fact checking website, more balanced and annotated to help the model train, this dataset will be used in the experiment. The claims are paired with news headlines and their stances and the public veracity of the claim. Both RumourEval 2019 and Emergent datasets are annotated in some way that aids in the training of our model, but the primary difference is that emergent claims data are longer and have multiple target topics.

In this experiment, Macro-averaged F1 is used to evaluate the performance on both tasks because it solves the imbalanced data problem. We also evaluate the proposed framework using an additional dataset, Emergent corpus, since headline annotations draw attention to the article where Emergent is a dataset of rumours (claims) coupled with news headlines and their stances. Since our model focuses on generating target-specific based claim s for news articles and summarising a long article into a target-specific based claim, it uses extra information, like the headline in Emergent data that represents the news store, to increase the accuracy.

9.3.2. Experimental Setting

In the experiments, the models in this paper are implemented using Keras. All word vectors are initialised using word2vec [140] and Glove [146], Where we discover that Glove performs better than word2vec. The hyperparameters, variables set before training which values are used to control the learning process and before optimizing the weights and bias, are chosen to achieve the most considerable value on the validation set and then train the model on the entire dataset. In our implementation, the word embedding dimension is 300, the size of hidden units in GRU is 100. The batch size is set 32. The learning rate is set at 0.001. The rule activation function used in the hidden layers is set to evaluate Task A and B's performance. The used evaluation metric: “macro-average F1 score” since the class labels are imbalanced.

9.3.3. Baseline Comparison

In this section, the performance of stance detection and rumour verification for the proposed model against the state-of-the-art model is discussed.

Emergent Dataset Training and Testing: Our model is compared against the state-of-the-art model reported in Zhang et al. [272] on the augmented Emergent dataset where the claims are more longer. In addition, experiments are performed on the publicly available Emergent dataset [257], consisting of news article headlines, evidence of news article content and stances. Stances can be classified as support, oppose, and discuss, which can infer veracity.

The results are shown in table 9.6. For veracity detection, our model obtains an accuracy of 78.83%. Since most previous models on the Emergent dataset focus only on the stance detection task but the veracity task, no comparison with the baseline is made, as shown in table 9.2.

Table 9.2 Performance comparison of the model against the State-of-the-Art model [272] for stance detection task on Emergent dataset

Model	Accuracy (%)		
	agree	disagree	discuss
Zhang et al's model [272]	82.52	69.05	84.30
Our model	**83.12**	**73.89**	**89.13**

RumourEval 2019 Dataset Training and Testing: Li et al.'s [96] and Khandelwal's [285] models show better performance compared with the top-5 systems in RumourEval 2019 [288]. Li et al.'s model [96], achieves the best performance for veracity detection, but they did not present results for stance detection as a single task. A comparison is made with the state of the art model by Khandelwal [285] as shown in table 9.3. For veracity checking, the comparison with Li et al. [96] as illustrated in table 9.4.

Table 9.3 Test results for Task A: stance detection on RumourEval

The model	**Macro-F**
Khandelwal's[285] Method – Top N_s using (A + B + C)	0.672
Our proposed model	**0.695**

Table 9.4 Test results for Task B: rumour veracity on RumourEval

The model	**Macro-F**
Li et al.'s model[96]	0.606
Our proposed model	**0.647**

For task A, stance detection, the work in Khandelwal [285] achieves the best Macro-f of 0.6720, and our model achieves 0.695, while for task B, veracity checking, the work in Li et al. [96] achieves the best Macro-f of 0.606 and our model achieves 0.647.

9.3.4. Ablation Study

By examining research question RQ2, ablation tests are conducted on the target topic extraction module, relevant clause retrieval and target-specific based claim generation, weighted ensemble learning for stance detection, and argumentation-based truth discovery; the results are shown in table 9.5 and table 9.6. As a result, each module significantly improves overall performance, demonstrating its efficacy.

Table 9.5 Ablation experiment of our model, stance detection scores of different ablation models.

Model	On emergent relative score	On RumourEval Macro-f
ATD without target topic extraction	82.19	0.623
ATD without target-specific based claim generation	79.36	0.679
ATD without clause relevant retrieving for target-specific based claim generation	83.92	0.678
ATD without evaluators for target-specific based claim generation	82.64	0.685
ATD without weighted ensemble learning for stance detection (i.e., each model contributes equally)	84.72	0.627
ATD without argumentation-based truth discovery	85.84	0.635
ATD with argumentation-based truth discovery	**89.97**	**0.695**

Table 9.6 Rumour veracity of scores of different ablation models.

Model	On emergent relative score	On RumourEval Macro-f
ATD without target topic extraction	75.77	0.603
ATD without target-specific based claim generation	72.08	0.639
ATD without clause relevant retrieving for target-specific based claim generation	73.83	0.636
ATD without evaluators for target-specific based claim generation	74.83	0.631
ATD without weighted ensemble learning for stance detection (i.e., each model contributes equally)	72.83	0.576
ATD without argumentation-based truth discovery	71.83	0.583
ATD with argumentation-based truth discovery	**78.83**	**0.647**

9.4. Discussions and Analysis

From the results given above, it is clear that the proposed method shows the best performance among these models. Furthermore, the proposed model outperforms both tasks, achieving Macro F1 0.695 for Task A and 0.647 for task B of the RumourEval 2019 dataset. The remainder of the subsection provides an analysis and evaluation of the proposed model and the results. First, training this model with and without applying the first component: the primary claim target topic extraction. The performance results revealed that this guide focuses more on the claim's primary target topic and positively contributes to stance classification performance. Target topic-clause retrieving help to ignore noise information as the noise may give wrong indications to deceive the model. This model is trained to classify stances without considering the target topic information, and a decreased accuracy is obtained. To show that the stance and rumour detection benefit from target topic aware target-specific based claim, experiments are conducted to detect evidence against claims without making a target-specific based claim based on the target topic. The change of macro-F1 scores on the two datasets shows the improvements by capturing certain words related to the target topic and eliminating the irrelevant. The macro-F1 score is chosen as a metric to give each task equal weight because it resolves the data imbalance. It outperforms the previous best baseline methods for the Emergent data. This could be that the model detects the article's stance against a claim by paying more attention to the claimed target topic, while the original article may have various target topics to talk about it. It is observed that word alignment can capture the target topic information for better performance of stance detection as target topic-specific attention provides more concise information, discarding another target topic the claim does not concern with.

The results emerging from these experiments confirmed the effectiveness of generating target-specific based claim conditioned on the target topic representation that is finally presented to the target topic claim and showed that it could be useful by extracting salient information from a long article without including less salient information., A significant improvement in this model's general results on both tasks A and B is achieved. Compared with baselines for stance detection, the advantage of knowledgeable target topic and target-specific based claim is demonstrated. A significant improvement on an emergent dataset from 82.52 %, 69.05 %, 84.30 % in Zhang et al.’s model [272] , to 83.12%, 73.89%, 89.13% for the three stances labels respectively as illustrated in table 9.2.

To investigate the applicability of the proposed model on new unseen data, where there is no knowledge related to this event, truth discovery is very beneficial to generalise veracity prediction since it depends on estimation without supervision. Despite unobserved samples, they may have semantic and syntactic features to that unseen news. The proposed model works well for the different text from two different datasets.

The following observations have been made based on the ablation experiments, as in table 9.5 and table 9.6:

- For the Emergent data, the veracity detection accuracy decreases when the target-specific based claim generation is not considered. This is particularly the case for a long article since the headline captures the primary information in making first impressions to readers.
- When the generated target-specific based claim does not cover the target topic of the claim or the extracted target topic is not valid, the performance is decreased
- Models augmented with truth discovery perform better than those without, i.e., assigning more scores to the claims inferred by more trusty sources.
- A significant improvement in integrating both tasks stance detection with rumour prediction.
- Since the sources' trustworthiness is not available and there is no prior information, this work's method can significantly enhance reliability source inference by estimating the trust based on Argumentation-based Truth Discovery.
- Utilising the claimed target topic helps the generator produce a concise target-specific based claim, and the evaluator can narrow the cosine distance.
- unlike that most of the previous studies as discussed earlier that either detect stance detection without considering the target topic or focused on inferring the stance for a set of predefined target topics [444], our model extracts a specific target topic to predict the stance toward it separately from stance classification,
- For size limitation, deep learning models need a high volume of data for training; it requires larger datasets than currently available, so this model is expected to perform better if more samples are obtained.
- Sometimes the e model fails to predict some stance labels correctly, maybe due to the lack of current information and other external evidence, e.g., the warrant is needed, so merging them may make additional enhancements, especially in the case.

9.5. Summary

A multi-task learning framework for jointly predicting rumour stance and veracity is proposed, where the source reliability is considered. A new deep learning model with a novel architecture is designed and studied to discover multiple truths from conflicting sources by connecting truth discovery methods with bipolar argumentation. The experiments with two influential datasets show that the proposed model outperforms state-of-the-art stance detection and rumour verification tasks. Argumentation-based truth discovery provides an effective way towards veracity detection by discovering the acceptable arguments through reframing truth discovery in terms of argumentation; this implies describing the arguments and the attack and support relations.

Chapter Ten: **Conclusion and Future Work**

10.1. Conclusion

This thesis examined the concept of false information detection in the news domain. We have concentrated our efforts on a single scenario that shaped the direction of the presented research: the possibility of improving the performance of existing models. Our main objective, as stated in the introduction, was to improve the performance of deep learning models for false information detection. For example, deep learning-based false information detection requires a large amount of data to verify claims and improve performance, but the information available is extremely limited. As a result of this void, the thesis was extended to address this challenge.

To accomplish the aforementioned goal, we first focused on verifying emerging claims when no reliable resources are available through style analysis and then trained the model to generalise to previously unseen data, as illustrated in chapter 4. Concerning the second method of verifying claims where reliable evidence is insufficient to detect the claim's validity, we proposed to consider warrant as additional information to justify the relationship, as discussed in chapter 5, and to account for the scarcity of warrant data, as discussed in chapter 6. In chapter 7, we demonstrate how generating correct claims, that explain why the decision of factuality is taken, from reliable evidence can assist in obtaining better results by comparing concise generated claims to user claims. Additionally, where emerging claims are accompanied by divergent posts from users without reliable evidence, utilising stances to ascertain the claim's veracity may be beneficial. Thus, we discuss the issue of scarcity of generated perspectives from other users and by incorporating common-sense knowledge, we propose a hybrid model that utilises reinforcement learning and multiple-head attention. We train the model on each stage used, as illustrated in chapter 8. Finally, in chapter 9, we developed a model to unify the two most common tasks for detecting false information, stance detection and claim verification.

In section 1.2, there are a variety of research questions around false information detection that have been examined. The remainder of this chapter describes how well this thesis' study addressed these research issues and summarises the contributions.

As previously discussed in Chapter 4, this chapter contributes the following: To detect fake information, we propose a novel hybrid architecture that combines multiple deep learning models to automatically extract salient deep characteristics and incorporates hand-crafted features.

Unlike previous work that relied on hand-crafted features or applied one of the deep learning models to discover the most important features, our proposed model can learn domain-invariant features of false information across multiple domains by leveraging the power of both critical features for prediction and features that ensure the model's performance.

Experiments on two real-world datasets demonstrate that the proposed method significantly improves false news detection performance by leveraging multiple deep learning models capable of accurately representing news syntactically and semantically and leveraging linguistic features, outperforming state-of-the-art algorithms.

This thesis addressed numerous obstacles for detecting fake information, beginning in chapter 4 with the absence of authoritative evidence or a set of replays' posts correlating to the claim. This occurs when a new claim is released but does not go widespread, allowing it to be recognised and presented by experts, and when individuals do not interact with the claim, expressing their ideas and sharing pertinent information. Given the reliable evidence, a claim is true if the evidence supports it; it is false if the evidence contradicts it. If no evidence is available but numerous replays of this claim exist, they could be utilised to provide context for the claim. Due to the diverse domain from which fake news originates, present models are still ineffective, and performance should be improved. While deep learning models are capable of representing text without hand-crafted features, we suggest that they benefit from them because they give auxiliary information that aids in model prediction. We argue that rather than using individual models, we should combine many deep learning models to achieve a more natural semantic and syntactic representation for learning domain-invariant features.

Chapter 4 examines the role of news style in attracting users, claiming that those who spread false information frequently employ scare stories, exaggeration, and fantastic language-based or opinions to try to get attention and help convince users to interact with the false news. We discover that style has a significant effect on how text affects a reader. Our findings contribute to a better understanding of the written style's effectiveness in detecting false information when hyprid deep learning is used. To summarise, when news articles are unseen and emerging without established knowledge bases, their credibility is determined by their writing style, such as linguistic cues, rather than their content. Our experiments demonstrated that style analysis outperforms the state of the art on two corpora with gold standard annotations, Twitter and news. Additionally, we discovered that they outperform other deep learning models statistically. As a result, our initial hypothesis H1 was confirmed.

In chapter 5 of this thesis, we examine the efficacy of leveraging warrants (which explain why the evidence implies the claim) for improving fact checking performance given an argument, a claim, and evidence. Our model employs Hierarchical Reinforcement Learning to select the most plausible warrant from annotated warrant data and Multi-Channel Multi-Head Attention to represent the input, combining all the represented inputs: the beast warrant, claim, and evidence to reach a factuality decision. Our experiments demonstrate that utilising warrants can significantly improve fact checking performance.

As expressed in our second hypothesis, H2, we hypothesise that using other knowledge such as to warrant A Novel Model for Enhancing Fact Checking: To tackle the fact checking insufficient evidence problem, we propose a model that uses extra information as a warrant to alleviate the problem. When there is insufficient reliable evidence and an ambiguous justification for supporting or attacking, we discuss how to improve the performance of current models, which necessitates recognising the implicit link between a claim and a piece of evidence (i.e., warrant). For fact checking tasks, we investigate the effectiveness of automatically executing warrants. While the warrant is frequently implied in the article to demonstrate how the evidence logically supports the claim, we discover that an explicit warrant clarifies the question more concisely, how did you arrive? by demonstrating how the evidence supports your claim as evidence for the fact. The findings indicate that using a warrant to justify the relationship between the claim and the evidence can assist in determining whether the claim is true or not. We thus confirmed H2.

Chapter 6 of this thesis presents the first study on automatically generating warrants that effectively describe the reasoning behind a factual prediction. We examine and compare various NLP techniques, such as RST and causality, in order to generate high-quality explanations. We demonstrate that pre-trained language combined with multi-agent reinforcement learning can significantly improve the performance of our baseline warrant generation models, resulting in warrants with greater diversity and quality.

The lack of labelled data limits the application of deep learning techniques to a wide range of tasks, such as fact checking, and contributes to the fact that existing fact checking models do not generalise well to new data. Developing a model to generate data can help alleviate labelled data scarcity by increasing the size of training data. The scarcity of labelled data for warrants must be addressed to maximise the benefits of deep learning algorithms for false information detection. To address the scarcity of labelled warrants data, this chapter proposes new models.

We discuss several Deep Learning models for generating Toulmin Argument warrants, including reinforcement learning, multi-head attention, pre-trained language models, and multi-agent systems. We showed how each of these models performed and how combining them with a reinforcement learning agent improved generation and inference accuracy. Our findings show that auxiliary data such as topic and sentiment must be combined with our model. By including a reinforcement learning agent, the generator can receive rapid and consistent training for successfully decoding sequential text. We get the best results on the dataset based on using pre-trained language model with multi-agents reinforcement network. The remaining Toulmin Arguments: supporting evidence, modifiers, and rebuttals will receive additional attention in future works.

We develop two models for generating factual claims in chapter 7: either by editing the claim to be checked in order to generate a new modified claim guided by the evidence, or by using the modifier to generate a new claim based on the evidence (generator). Our models overcome the scarcity of data problem caused by a lack of datasets containing claims and their corrections by repurposing the available dataset without using any external data. Experiments demonstrate that our model is capable of correcting factual errors and optimising the fact checking task's performance.

We develop neural network-based models that make a factual claim based on claim context information in the case of a non-factual claim. We discover that the neural network-based model performs better when it is used to modify misleading information rather than when it is used to generate a new claim from its premises. We investigated the problem of encoding lengthy evidence articles to generate a factual claim and demonstrated that using hierarchical reinforcement learning could improve generation by automatic evaluation. The analysis demonstrates that this improvement is due to the multi-ability agent's covering all relevant claim information and generate a factual claim. For claim modification models, a sequence operation-based method combined with hierarchical reinforcement learning effectively addresses the problem of untrue claims.

In chapter 8, we propose a novel approach for a novel task: automatically generating diverse and multiple arguments in response to claim about a contentious topic. We demonstrate how utilising common-sense knowledge as auxiliary information prevents generated perspectives from contradicting well-known facts. The experimental results demonstrate that the new model outperforms all previous benchmarks in terms of automatic metrics, quality, and diversity. Our additions are specifically aimed towards reducing the bias in the training data, by having a false claim appear in both “Agrees” and “Disagrees” classes. Generation of synthetic claim-evidence pairs to augment an existing biased fact checking dataset in order to improve the performance of trained classifiers on an unbiased dataset. As mentioned, our datasets were mostly imbalanced, but the current models did not consider the class imbalance. This problem causes the model to bias toward the majority class.

Depending only on a single view of textual information without considering other users and sources, metaknowledge is not always sufficient. Also, some of the information sources are completely unreliable. One challenge to this task is evaluating the trustworthiness of the website content. Deciding the credibility of claims based on one view may not be sufficient, and an implicit bias is a likely present. So that we are required to consider all sides of this issue (perspectives from diverse sources). To eliminate the biases and the scarcity of relevant perspectives, a model needs to generate a diverse range of perspectives from trustworthy sources to evaluate the claim. Because new claims may lack reliable evidence, considering other users' perspectives can aid in determining whether or not a claim is true. As a first step toward their use in determining the veracity of claims.

To alleviate the scarcity of labelled data and bias problems, we proposed several models to generate claim perspectives with opposite stances and generate warrant that links claims to evidence. The generated perspectives could be used to retrieve evidence documents in the conflict information cases, then viewing the claim from divers' sources eliminates bias. Regarding warrant generation, we suppose that the possibility of linking the claim with evidence means support relation else either refute or unverified. This helps increase the models' accuracy, which only has only two inputs in its model, claim and evidence. So that warrant help model to a better understanding with an interpretable explanation to decide the relation. Each agent learns to facilitate learning independently, with its actor, critic, observation, and actions, without sharing or communicating with other agents. Multiple attitudes and adversarial claims will be generated for each claim argument to obtain more diverse information and create robust adversarial samples, where each perspective takes a position on a claim based on evidence text. To help the reader perceive more debatable claims and determine their veracity, information should be gathered from a diverse yet comprehensive set of perspectives based on the claim argument. Multiple views with the same or opposite meaning could be used to express the claim. If the perspective has disputed or refuted the claim, it should be determined whether or not information exists to substantiate this contentious issue. A common-sense knowledge can be used to improve our model. Results show that the quality and diversity of the generated perspectives can be effectively increased common-sense knowledge method.

As previously discussed in chapter 9, this chapter makes the following contributions: We propose a novel framework for tackling stance classification and veracity checking concurrently. As far as we are aware, this is the first work to employ argumentation-based truth discovery. A novel model for the optimal target-specific based claim generator is proposed for the lengthy rumour with multiple target topics while keeping the document's primary target topic in mind. Target topic extraction enables the examination of all pertinent arguments to determine the truth by the same target topic. The first model to integrate fact checking, stance detection, and truth discovery. The application of the Emergent and SemEval 2019 datasets demonstrates that this framework outperforms existing methods in both stance classification and veracity checking.

The current approaches for false information detection analyse each social media post as a single unit and are limited to individual sources of claims instead of considering the context surrounding them. It is necessary to investigate how to automatically predict the veracity of rumours spread on social media by analysing other perspectives' stances. After assessing people's reactions to something, another challenge is how we can effectively determine what is likely to be true. One of the main problems with fact checking; is that the assessment of the truth of claim differs from reader to reader based on specialist knowledge of the subject matter and linguistic knowledge in addition to the level of experience with other characteristics

Perspectives are other users' attitudes toward supporting or opposing a claim based on evidence sources, which means that multiple arguments or counterarguments may be associated with the same claim. This implies that, particularly concerning contentious issues, the reader should be aware of and accept what others say, taking into account other supported or refuted perspectives. We require a stance detection task to identify alternative perspectives on the claims. Analyzing contentious claims from multiple perspectives enables the development of alternative interpretations and a more precise and thorough understanding [445].

Integrating stance detection and factuality checking: to address the problem of detecting the false information on conflict sources that contributes equally as independent tasks, we propose a novel model that combines stance detection with fact checking to determine the factuality by aggregating documents' stances. We empirically study the unified model's performance concerning baselines methods, and the results show that the model performs better in terms of F-score.

In experiments, integrating stance detection and fact checking improved performance. Additionally, on these tasks, our models outperformed baselines. The positions users take on claims can help predict their veracity. We then discuss how to combine stance predictions and fact checking based on conversational stances. The results show that the veracity of a claim can be determined by aggregating the strength of the stances, without requiring information about specific users, assuming that all sources contribute equally.

Multi-Task Learning Framework for Stance Detection and Veracity Prediction addresses the problem of detecting false information on conflict sources with variant reliability sources as independent tasks. This work translates truth discovery to argumentation framework to solve the conflict information. We incorporate the stance information in the model, and for the lengthy document, we generate a conclusion to capture the more relevant information for the critical aspect of the claim. A few studies have taken into account contextual information. The source reliability is taken into account in a multi-task learning framework for jointly predicting rumour stance and veracity. By combining truth discovery methods with bipolar argumentation, a new deep learning model with a novel architecture is designed and studied to discover multiple truths from conflicting sources. The proposed model outperforms state-of-the-art stance detection and rumour verification tasks in experiments with two influential datasets. Argumentation-based truth discovery is a useful method for detecting veracity by identifying acceptable arguments by reframing truth discovery in terms of argumentation; this entails describing the arguments as well as the attack and support relations.

10.2. Future Work

Although our models perform well and outperform state-of-the-art methods on standard benchmarks, it would be interesting to investigate additional issues in the future, including the use of large and complex datasets, consideration of languages other than English, and development of language-specific fact checking methods, in addition to exploring other directions for each of our models. We concentrated on textual information in this thesis; however, in the future, we intend to consider other forms of misleading information, such as images and ensure that the image is semantically consistent with the surrounding text. We intend to retrieve evidence online rather than from datasets, particularly in domains where facts are subject to change, such as the COVID-19 pandemic information, for which no gold standard exists at the moment. Additionally, by incorporating evidence retrieval, we can obtain sufficient and large data to constrain the generalisation problem of our models.

While the chosen embedding model in chapter 4 is effective for our Cross-Domain Factuality Checking model, future research will incorporate a more sophisticated language representation model, such as BERT. As the existing dataset is primarily domain-specific, it constrains the performance of fact checking models. In the future, a more diversified dataset that incorporates diverse features such as emotions that may be valuable will be developed, along with an increase in the number of labels such as "mostly true" or special labels for identifying specific types of fake news.

Due to the fact that we lack sufficient labelled data, we must rely on feature engineering, which is laborious and time consuming; therefore, increased labelled data could aid deep learning models in producing better text representation without the use of additional features. Certain untrustworthy evidence may be presented in a credible manner and then mislead the reader into believing the claim is true; thus, developing methods for determining the credibility of sources and incorporating the source credibility of a claim into our model may be beneficial, as it may aid in the early detection and eradication of fake news. Addressing this issue could be accomplished in a variety of ways, including determining the extent to which sources corroborate established facts or the extent to which sources agree with well-known sources of evidence.

It would like to integrate our model with techniques that are frequently used in the news domain for highly related tasks, such as uncertainty identification, stance detection, and sentiment detection, in future work.

We will devote additional attention to the remaining Toulmin Arguments for future works as in chapter 5: supporting evidence, modifiers, and rebuttals. We will look for additional reasons to support Backing, based on examples and sentences that contain the words prove, means, show, confirmed, and others. Data pertaining to a claim. For instance, secondary scored candidates warrant, considered as Backing, searching for sentences containing dictionary modifiers or containing contradiction relations (rebuttals).

While our warrant-aware fact checking has improved significantly, there is still a challenge arising from composite claims, where the selected warrant may link the claim to evidence regarding a specific aspect or target, but not all targets. For instance, if the minister claims that salaries are increased at the start of the new year and the weekend begins on Friday, the warrant may support one part but not both, confusing the model and resulting in an incorrect label decision. Thus, segmenting claims, for example, according to targets or aspects, may assist in resolving this issue. The development of multimodal fact checking systems will be facilitated by the creation of large-scale annotated datasets paired with evidence beyond metadata.

While increasing and improving the quality of generated warrants remains a research objective, developing systems that identify relevant warrants from the online web or from a large warrant annotated data set may aid in performance improvement.

We will continue to work on improving the quality and diversity of the warrants generated, as well as integrating the warrant generation mechanism into fact checking models, allowing it to be used to clarify any classification decision regarding factuality.

Additional research and development of more sophisticated models for reducing available false information and analysing its impact across multiple domains are required for factual claim generation. Additionally, we will examine people's willingness to accept corrected information and the possibility of reversing their decision.

We believe that in the future, factual claim correction should be prioritised as a means of convincing the user of the robustness of the fact checking decision and of succinctly capturing misleading information that is inconsistent with trustworthy evidence.

As we can see, generating perspectives remains a challenging problem worth investigating. Other language generation models, in particular those that improved the quality and diversity of generated perspectives and achieved state-of-the-art performance on a variety of relevant generating tasks, such as gpt-2, could be used as enhanced decoding methods. In the future, we could use the outputs of the generated perspectives to augment the imbalance data, thereby reducing the bias issue that limits the model's ability to generalise and improve performance.

We also intend to use a large and complex dataset in the future, which will include additional languages. Emoticons are frequently used in social media to represent reactions, and their inclusion would be critical for stance detection. Extending the modelling of user status is one of our study's future directions.

There are several ways to move the current work forward. The current work involves source-claim and source-source relationships and focuses on information richness to obtain confident score information. We are also planning to modify this model by considering other argumentation components such as warrants and backings in the Toulmin model and consider other factors like the source reputations.

We can also examine the applicability of alternative argumentation models such as abstract argumentation Rogerian argument, and the Classical or Aristotelian argument. Additionally, warrants, rebuttals, and backing may be considered to reinforce the veracity of our proposed model. Additionally, knowledge bases with an ontology structure, such as DBPedia (which can be queried using SPARQL), may serve as a source of facts that aid in veracity checking. For instance, if a lengthy claim contains portions that contradict well-established facts in DBPedia, the entire claim may be more likely to be false.

References

[1] H. Rashkin et al., ‘Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking’, in *Proceedings ofthe 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2931–2937 [Online]. Available: 10.18653/v1/d17-1317.

[2] J. Corner, ‘Fake News, Post-truth and Media–political Change’, *Media, Cult. Soc.*, vol. 39, no. 7, pp. 1100–1107, 2017 [Online]. Available: 10.1177/0163443717726743.

[3] S. Chen et al., ‘Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims’, in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019, vol. 1, pp. 542–557 [Online]. Available: 10.18653/v1/n19-1053.

[4] B. Riedel et al., ‘A simple but tough-to-beat baseline for the Fake News Challenge stance detection task’, *arXiv Prepr. arXiv1707.03264*, pp. 1–6, 2018[Online]. Availablehttps://arxiv.org/pdf/1707.03264.pdf.

[5] S. Baird et al., (2017,), *Talos Targets Disinformation with Fake News Challenge Victory*, cisco - BY WILLIAM LARGENT. [Online]. Available: https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html. [Accessed: 11 Nov. 2020].

[6] M. Mohtarami et al., ‘Automatic Stance Detection Using End-to-End Memory Networks’, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, vol. 1, pp. 767–776 [Online]. Available: 10.18653/v1/N18-1070.

[7] R. Rudinger et al., ‘Neural models of factuality’, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 731–744[Online]. Availablehttps://www.aclweb.org/anthology/N18-1067.

[8] J. 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, (Long Papers)*, 2018, vol. 1, no. June, pp. 809–819[Online]. Availablehttps://www.aclweb.org/anthology/N18-1074.

[9] A. Vlachos and S. Riedel, ‘Fact Checking: Task definition and dataset construction’, in *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, 2014, pp. 18–22 [Online]. Available: 10.3115/v1/w14-2508.

[10] J. Ma et al., ‘Rumor Detection on Twitter with Tree-structured Recursive Neural Networks’, in *Proceedings ofthe 56th Annual Meeting ofthe Association for Computational Linguistics (Long Papers)*, 2018, pp. 1980–1989[Online]. Availablehttps://www.aclweb.org/anthology/P18-1184.

[11] J. MA et al., ‘Detecting Rumors from Microblogs with Recurrent Neural Networks’, in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016, pp. 3818–3824[Online]. Availablehttps://ink.library.smu.edu.sg/sis_research/4630.

[12] G. Bhatt et al., ‘On the Benefit of Combining Neural, Statistical and External

Features for Fake News Identification', in *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, 2018, pp. 1353–1357 [Online]. Available: 10.1145/3184558.3191577.

[13] Z. Qian et al., 'Event Factuality Identification via Generative Adversarial Networks with Auxiliary Classification', in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018, vol. 2018-July, pp. 4293–4300 [Online]. Available: 10.24963/ijcai.2018/597.

[14] A. Hanselowski et al., 'UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification', in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2019, no. November, pp. 103–108[Online]. Availablehttps://www.aclweb.org/anthology/W18-5516.

[15] H. Kilicoglu et al., 'Assigning Factuality Values to Semantic Relations Extracted from Biomedical Research Literature', *PLoS One*, vol. 12, no. 7, pp. 1–20, 2017 [Online]. Available: 10.1371/journal.pone.0179926.

[16] R. Bar-Haim et al., 'Stance Classification of Context-Dependent Claims', in *Proceedings ofthe 15th Conference ofthe European Chapter ofthe Association for Computational Linguistics: Long Papers*, 2017, vol. 1, no. April, pp. 251–261 [Online]. Available: 10.18653/v1/e17-1024.

[17] N. Ruchansky et al., 'CSI: A Hybrid Deep Model for Fake News Detection', in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, vol. Part F1318, no. November, pp. 797–806 [Online]. Available: 10.1145/3132847.3132877.

[18] J. Zhang et al., 'FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network', *Proc. Int. Conf. Data Eng.*, vol. April, pp. 1826–1829, 2020 [Online]. Available: 10.1109/ICDE48307.2020.00180.

[19] Y. Yang et al., 'TI-CNN: Convolutional Neural Networks for Fake News Detection', *CoRR*, vol. abs/1806.0, 2018[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1806.html#abs-1806-00749.

[20] Y. Wang et al., 'EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection', in *Proceedings of The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, vol. Article 4, pp. 849–857[Online]. Availablehttps://doi.org/10.1145/3219819.3219903.

[21] G. Gravanis et al., 'Behind the Cues: A Benchmarking Study for Fake News Detection', *Expert Syst. Appl.*, vol. 128, pp. 201–213, 2019[Online]. Availablehttps://doi.org/10.1016/j.eswa.2019.03.036.

[22] C. Cayrol and M. C. Lagasquie-Schiex, *Gradual Valuation for Bipolar Argumentation Frameworks*, vol. 3571 LNAI, no. June. Berlin, Heidelberg: Springer, Berlin, Heidelberg, 2005[Online]. Availablehttps://doi.org/10.1007/11518655_32.

[23] S. KUMAR and N. SHAH, 'False Information onWeb and Social Media: A Survey', *arXiv*, vol. 1, no. 1, 2018.

[24] S. F. Aikin, 'Poe's Law, Group Polarization, and the Epistemology of Online Religious Discourse', *SSRN*, no. January, pp. 301–317, 2009[Online]. Availablehttps://papers.ssrn.com/sol3/papers.cfm?abstract_id=1332169.

[25] D. Fallis, 'A Conceptual Analysis of Disinformation', in *Proceeding of iConference*, 2009[Online]. Availablehttps://www.ideals.illinois.edu/handle/2142/15205.

[26] A. Smith and V. Banic, (2016,), *Fake News: How a Partying Macedonian Teen*

Earns Thousands Publishing Lies, NBC News. [Online]. Available: https://www.nbcnews.com/news/world/fake-news-how-partying-macedonian-teen-earns-thousands-publishing-lies-n692451. [Accessed: 08 Dec. 2020].

[27] C. Budak et al., ‘Limiting the Spread of Misinformation in Social Networks’, in *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, 2011, no. March 28-April 1, pp. 665–674 [Online]. Available: 10.1145/1963405.1963499.

[28] M. C. Forelle et al., ‘Political Bots and the Manipulation of Public Opinion in Venezuela’, *SSRN Electron. J.*, pp. 1–8, 2015 [Online]. Available: 10.2139/ssrn.2635800.

[29] P. N. Howard and B. Kollanyi, ‘Bots, #Strongerin, and #Brexit: Computational Propaganda During the UK-EU Referendum’, *SSRN Electron. J.*, pp. 1–6, 2017[Online]. Availablehttps://ssrn.com/abstract=2798311.

[30] C. Shao et al., ‘The Spread of Low-Credibility Content by Social Bots’, *Nat. Commun.*, vol. 9, no. 1, pp. 1–9, 2018[Online]. Availablehttp://dx.doi.org/10.1038/s41467-018-06930-7.

[31] G. S. Jowett and V. O’Donnell, *Propaganda and Persuasion*, 7th-editio ed., no. Chapter 4. 2014.

[32] V. L. Rubin et al., ‘Towards News Verification: Deception Detection Methods for News Discourse’, in *Proceedings of the Hawaii International Conference on System Sciences (HICSS48) Symposium on Rapid Screening Technologies*, 2015, no. January, pp. 1–11[Online]. Availablehttps://works.bepress.com/victoriarubin/6/.

[33] X. Chen et al., ‘Why Do Social Media Users Share Misinformation?’, in *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2015, pp. 111–114[Online]. Availablehttps://doi.org/10.1145/2756406.2756941.

[34] M. Fenster, *Excerpt from Conspiracy Theories: Secrecy and Power in American Culture*, Revised ed., no. January. University of Minnesota Press, 2008[Online]. Availablehttps://works.bepress.com/mark_fenster/11/.

[35] S. Kumar et al., ‘Disinformation on the Web: Impact, Characteristics, and Detection of Wikipedia Hoaxes’, in *25th International World Wide Web Conference, WWW 2016*, 2016, pp. 591–602[Online]. Availablehttp://dx.doi.org/10.1145/2872427.2883085.

[36] M. Potthast et al., ‘A Stylometric Inquiry into Hyperpartisan and Fake News’, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2018, vol. 1, no. July, pp. 231–240[Online]. Availablehttps://www.aclweb.org/anthology/P18-1022.

[37] L. Feldman, ‘Partisan Differences in Opinionated News Perceptions: A Test of the Hostile Media Effect’, *Polit. Behav.*, vol. 33, no. 3, pp. 407–432, 2011 [Online]. Available: 10.1007/s11109-010-9139-4.

[38] W. A. Peterson and N. P. Gist, ‘Rumor and public opinion’, *Am. J. Sociol.*, vol. 57, no. 2, pp. 159–167, 2014[Online]. Availablehttps://www.jstor.org/stable/2772077?seq=1.

[39] W. J. Campbell, *Yellow Journalism: Puncturing the Myths, Defining the Legacies*. Westport, Conn. ; London : Praeger, 2001[Online]. Availablehttps://books.google.co.uk/books?hl=ar&lr=&id=-_kWbKnrx8AC&oi=fnd&pg=PP9&dq=Campbell,+W.+Joseph.+%22Yellow+Journalism:+Puncturing+the+Myths.%22+Defining+the+Legacies+(2001).&ots=NHU5GOUtJi&sig=l1IUJ2QzlES519sWmzQznB2geXQ&redir_esc=y#v=onepage&q=Campbell%252.

[40] B. Y. E. Ferrara et al., ‘The Rise of Social Bots’, *Commun. ACM*, vol. 59, no. 7, pp. 96–104, 2016[Online]. Availablehttps://doi.org/10.1145/2818717.

[41] V. S. Subrahmanian et al., ‘The DARPA Twitter Bot Challenge’, *Computer (Long. Beach. Calif).*, vol. 49, no. 6, pp. 38–46, 2016 [Online]. Available: 10.1109/MC.2016.183.

[42] P. N. Howard et al., ‘The IRA, Social Media and Political Polarization in the United States, 2012-2018’, *U.S. Senate Documents, Congress of the United States*. DigitalCommons@University of Nebraska - Lincoln, Nebraska - Lincoln, U S, pp. 2012–2018, 2019[Online]. Availablehttps://digitalcommons.unl.edu/senatedocs.

[43] Facebook, (2020,), *What's the difference between organic, paid and post reach?*, Facebook Help Centre. [Online]. Available: https://www.facebook.com/help/285625061456389. [Accessed: 01 Dec. 2020].

[44] P. Snyder et al., ‘Fifteen Minutes of Unwanted Fame: Detecting and Characterizing Doxing’, in *Proceedings of the 2017 Internet Measurement Conference*, 2017, vol. Part F1319, no. November 1-3, pp. 432–444[Online]. Availablehttps://doi.org/10.1145/3131365.3131385.

[45] M. Ott et al., ‘Finding Deceptive Opinion Spam by Any Stretch of the Imagination’, in *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, vol. 1, pp. 309–319[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1107.html#abs-1107-4557.

[46] H. Allcott and M. Gentzkow, ‘Social Media and Fake News in the 2016 Election’, *J. Econ. Perspect.*, vol. 31, no. 2, pp. 211–236, 2017[Online]. Availablehttps://doi.org/10.1257/jep.31.2.211.

[47] A. Friggeri et al., ‘Rumor Cascades’, *Proc. 8th AAAI Int. Conf. Weblogs Soc. Media, ICWSM 2014*, pp. 101–110, 2014.

[48] Craig Silverman, ‘This Analysis Shows How Viral Fake Election News Stories Outperformed Real News On Facebook’, *BuzzFeed News*, Toronto, pp. 1–7, Nov. 16, 2016[Online]. Availablehttps://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook.

[49] C. Silverman, ‘Lies, Damn Lies, and Viral Content’, 2015[Online]. Availablehttps://doi.org/10.7916/D8Q81RHH.

[50] M. Fisher et al., ‘Pizzagate: From rumor, to hashtag, to gunfire in D.C.’, *Washington Post*, p. Local 6, Dec. 06, 2016[Online]. Availablehttps://www.washingtonpost.com/local/pizzagate-from-rumor-to-hashtag-to-gunfire-in-dc/2016/12/06/4c7def50-bbd4-11e6-94ac-3d324840106c_story.html.

[51] K. Starbird et al., ‘Rumors, False Flags, and Digital Vigilantes: Misinformation on Twitter after the 2013 Boston Marathon Bombin’, in *iConference 2014 Proceedings*, 2014, pp. 654–662[Online]. Availablehttp://hdl.handle.net/2142/47257.

[52] A. Gupta et al., ‘Faking Sandy: characterizing and identifying fake images on Twitter during Hurricane Sandy’, in *WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 729–736[Online]. Availablehttps://doi.org/10.1145/2487788.2488033.

[53] J. Bollen et al., ‘Twitter Mood Predicts the Stock Market’, *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, 2011[Online]. Availablehttp://dx.doi.org/10.1016/j.jocs.2010.12.007.

[54] E. Higgins, (2016,), *Fake news is spiraling out of control – and it is up to all of us to stop it*, Politics. [Online]. Available: https://www.ibtimes.co.uk/fake-news-spiralling-out-control-it-all-us-stop-it-1596911. [Accessed: 01 Dec. 2020].

[55] J. Gillin, (2017,), *The More Outrageous, the Better: How Clickbait Ads Make Money for Fake News Sites*, PolitiFact. [Online]. Available: https://www.politifact.com/article/2017/oct/04/more-outrageous-better-how-clickbait-ads-make-mone/. [Accessed: 01 Dec. 2020].

[56] US House of Representatives Permanent Select Committee on Intelligence, (2018,), *Exposing Russia's Effort to Sow Discord Online: The Internet Research Agency and Advertisements*, Social Media Content. [Online]. Available: https://intelligence.house.gov/social-media-content/. [Accessed: 01 Dec. 2020].

[57] S. Zannettou et al., 'The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans', *J. Data Inf. Qual.*, vol. 11, no. 3, pp. 1–37, 2019[Online]. Availablehttps://doi.org/10.1145/3309699.

[58] Y. Boshmaf et al., 'The Socialbot Network: When Bots Socialize for Fame and Money', in *Proceedings of the 27th Annual Computer Security Applications Conference*, 2011, pp. 93–102 [Online]. Available: 10.1145/2076732.2076746.

[59] S. Al-Khateeb and N. Agarwal, 'Examining Botnet Behaviors for Propaganda Dissemination: A Case Study of ISIL's Beheading Videos-Based Propaganda', in *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, 2015, pp. 51–57 [Online]. Available: 10.1109/ICDMW.2015.41.

[60] M. Luo, 'How the N.R.A. Manipulates Gun Owners and the Media', *The New Yorker*, Aug. 11, 2017[Online]. Availablehttps://www.newyorker.com/news/news-desk/how-the-nra-manipulates-gun-owners-and-the-media.

[61] C. Timberg, 'Spreading fake news becomes standard practice for governments across the world', *The Washington Post*, Jul. 17, 2017[Online]. Availablehttps://www.washingtonpost.com/news/the-switch/wp/2017/07/17/spreading-fake-news-becomes-standard-practice-for-governments-across-the-world/.

[62] C. Chen et al., 'Battling the Internet Water Army: Detection of Hidden Paid Posters', in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2013*, 2013, pp. 116–120 [Online]. Available: 10.1145/2492517.2492637.

[63] S. Zannettou et al., 'Disinformation Warfare: Understanding State-sponsored Trolls on Twitter and their Influence on the Web', in *The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019*, 2019, pp. 218–226[Online]. Availablehttps://doi.org/10.1145/3308560.3316495.

[64] S. T. Lee, 'Lying to Tell the Truth: Journalists and the Social Context of Deception', *Mass Commun. Soc.*, vol. 7, no. 1, pp. 97–120, 2004 [Online]. Available: 10.1207/s15327825mcs0701_7.

[65] RationalWiki, (2020,), *Useful idiot*, RationalWiki. [Online]. Available: https://rationalwiki.org/wiki/Useful_idiot. [Accessed: 06 Feb. 2021].

[66] S. Hook, (2020,), *Sandy Hook Elementary School shooting conspiracy theories*, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Sandy_Hook_Elementary_School_shooting_conspiracy_theories. [Accessed: 13 Jan. 2021].

[67] T. Mihaylov et al., 'Finding Opinion Manipulation Trolls in News Community Forums', in *Proceedings of the Nineteenth Conference on Computational Natural Language Learning-CoNLL*, 2015, pp. 310–314[Online].

Availablehttps://www.aclweb.org/anthology/K15-1032.

[68] N. Shah et al., 'EdgeCentric: Anomaly Detection in Edge-Attributed Networks', in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, vol. 0, pp. 327–334 [Online]. Available: 10.1109/ICDMW.2016.0053.

[69] V. Pérez-Rosas et al., 'Automatic Detection of Fake News', in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3391–3401[Online]. Availablehttps://www.aclweb.org/anthology/C18-1287.

[70] B. D. Horne and S. Adali, 'This Just In-Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News', in *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*, 2017, vol. 11, no. 1, pp. 40–49[Online]. Availablehttps://ojs.aaai.org/index.php/ICWSM/article/view/14976.

[71] V. Rubin et al., 'Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News', in *Proceedings of the second workshop on computational approaches to deception detection- NAACL-HLT*, 2016, pp. 7–17[Online]. Availablehttps://www.aclweb.org/anthology/W16-0802.

[72] T. Mitra et al., 'A Parsimonious Language Model of Social Media Credibility Across Disparate Events', in *CSCW '17: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 126–145[Online]. Availablehttps://doi.org/10.1145/2998181.2998351%0A.

[73] A. Bessi and E. Ferrara, 'Social Bots Distort the 2016 US Presidential Election Online Discussion', *Online Discuss.*, vol. 21, no. 11, pp. 1–15, 2016[Online]. Availablehttps://ssrn.com/abstract=2982233.

[74] F. Ma et al., 'FaitCrowd: Fine Grained Truth Discovery for Crowdsourced Data Aggregation', in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, vol. 2015-Augus, pp. 745–754[Online]. Availablehttps://ink.library.smu.edu.sg/sis_research/3258%0AThis.

[75] N. Nakashole and T. M. Mitchell, 'Language-Aware Truth Assessment of Fact Candidates', in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, 2014, vol. 1, pp. 1009–1019[Online]. Availablehttp://dblp.uni-trier.de/db/conf/acl/acl2014-1.html#NakasholeM14.

[76] J. Pasternack and D. Roth, 'Generalized Fact-Finding', in *Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011*, 2011, vol. 1, pp. 99–100 [Online]. Available: 10.1145/1963192.1963243.

[77] S. Mukherjee and G. Weikum, 'Leveraging Joint Interactions for Credibility Analysis in News Communities', *Proc. 24th ACM Int. Conf. Inf. Knowl. Manag.*, vol. 19-23-Oct-, pp. 353–362, 2015[Online]. Availablehttps://doi.org/10.1145/2806416.2806537.

[78] M. Lippi and P. Torroni, 'Argumentation Mining: State of the Art and Emerging Trends', *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 1–25, 2016[Online]. Availablehttps://doi.org/10.1145/2850417.

[79] A. Lytos et al., 'The evolution of argumentation mining: From models to social media and emerging tools', *Inf. Process. Manag.*, vol. 56, no. 6, p. 102055, 2019[Online]. Availablehttps://doi.org/10.1016/j.ipm.2019.102055.

[80] J. Ma et al., 'Detect Rumors on Twitter by Promoting Information Campaigns with Generative Adversarial Learning', in *Proceedings of the Web Conference (WWW 2019)*, 2019, pp. 3049–3055[Online]. Availablehttps://ink.library.smu.edu.sg/sis_research/4559.

[81] H. Guo et al., ‘Rumor Detection with Hierarchical Social Attention Network’, in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, no. October, pp. 943–952 [Online]. Available: 10.1145/3269206.3271709.

[82] F. Monti et al., ‘Fake News Detection on Social Media using Geometric Deep Learning’, in *arXiv preprint arXiv:1902.06673*, 2019.

[83] M. Dong et al., ‘Multiple Rumor Source Detection with Graph Convolutional Networks’, in *CIKM ’19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 569–578[Online]. Availablehttps://doi.org/10.1145/3357384.3357994.

[84] G. Karadzhov et al., ‘We Built a Fake News & Click-bait Filter: What Happened Next Will Blow Your Mind!’, in *Proceedings of Recent Advances in Natural Language Processing*, 2017, vol. September, pp. 334–343 [Online]. Available: 10.26615/978-954-452-049-6_045.

[85] G. Karadzhov et al., ‘Fully Automated Fact Checking Using External Sources’, *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, vol. 2017-Septe, pp. 344–353, 2017 [Online]. Available: 10.26615/978-954-452-049-6-046.

[86] M. Seyyed et al., ‘A Novel Text Mining Approach Based on TF-IDF and Support Vector Machine for News Classification’, *2nd IEEE Int. Conf. Eng. Technol.*, no. March, pp. 16–20, 2016[Online]. Availablehttps://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7569223.

[87] Z. Jin et al., ‘News Verification by Exploiting Conflicting Social Viewpoints in Microblogs’, in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 2972–2978.

[88] E. Tacchini et al., ‘Some Like it Hoax: Automated Fake News Detection in Social Networks’, *Proc. Second Work. Data Sci. Soc. Good*, vol. 1960, pp. 1–12, 2017.

[89] H. Ahmed et al., ‘Detecting Opinion Spams and Fake News Using Text Classification’, *Secur. Priv.*, vol. 1, no. 1, p. e9, 2018 [Online]. Available: 10.1002/spy2.9.

[90] Y. Long et al., ‘Fake News Detection Through Multi-Perspective Speaker Profiles’, in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, vol. Volume 2:, no. 8, pp. 252–256[Online]. Availablehttp://www.aclweb.org/anthology/I17-2043.

[91] Z. Yang et al., ‘Hierarchical Attention Networks for Document Classification’, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489[Online]. Availablehttps://www.aclweb.org/anthology/N16-1174.

[92] G. B. Guacho et al., ‘Semi-supervised Content-based Fake News Detection using Tensor Embeddings and Label Propagation’, in *Proc. SoCal NLP Symposium*, 2018, pp. 3–5[Online]. Availablehttps://www.cs.ucr.edu/~epapalex/papers/socal-nlp18.pdf.

[93] M. Hardalov et al., ‘In search of credible news’, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9883 LNAI, pp. 172–180, 2016 [Online]. Available: 10.1007/978-3-319-44748-317.

[94] F. A. Ozbay and B. Alatas, ‘A Novel Approach for Detection of Fake News on Social Media Using Metaheuristic Optimization Algorithms’, *Elektron. ir Elektrotechnika*, vol. 25, no. 4, pp. 62–67, 2019[Online]. Availablehttps://doi.org/10.5755/j01.eie.25.4.23972.

[95] A. Hanselowski et al., ‘A Retrospective Analysis of the Fake News Challenge Stance Detection Task’, in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1859–1874[Online]. Availablehttps://www.aclweb.org/anthology/C18-1158.

[96] Q. Li et al., ‘Rumor Detection By Exploiting User Credibility Information, Attention and Multi-task Learning’, in *Proceedings ofthe 57th Annual Meeting ofthe Association for Computational Linguistics*, 2020, pp. 1173–1179 [Online]. Available: 10.18653/v1/p19-1113.

[97] M. R. Morris et al., ‘Tweeting is Believing?: Understanding Microblog Credibility Perceptions’, in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 2012, no. 2, pp. 441–450[Online]. Availablehttps://doi.org/10.1145/2145204.2145274.

[98] K. Shu et al., ‘Understanding User Profiles on Social Media for Fake News Detection’, in *Proceedings - IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018*, 2018, no. March, pp. 430–435 [Online]. Available: 10.1109/MIPR.2018.00092.

[99] J. Cao et al., ‘Automatic rumor detection on microblogs: A Survey’, *arXiv Prepr. arXiv*, pp. 1–14, 2018[Online]. Availablehttps://arxiv.org/pdf/1807.03505.pdf.

[100] W. Fedus et al., ‘MaskGAN: Better Text Generation via Filling in the______’, in *arXiv preprint arXiv:1801.07736*, 2018, vol. abs/1801.0, pp. 1–17[Online]. Availablehttps://arxiv.org/pdf/1801.07736.pdf?source=post_page---------------------------.

[101] Y. Bengio et al., ‘A Neural Probabilistic Language Model’, *J. Mach. Learn. Res. JMLR*, vol. 3, no. February, pp. 1137–1155, 2003[Online]. Availablehttps://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf.

[102] J. Zhang et al., ‘Fake News Detection with Deep Diffusive Network Model’, *CoRR*, vol. abs/1805.0, no. May, pp. 1–11, 2018[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1805.html#abs-1805-08751.

[103] C. Castillo et al., ‘Information Credibility on Twitter’, in *Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011*, 2011, pp. 675–684[Online]. Availablehttps://doi.org/10.1145/1963405.1963500.

[104] C. Castillo et al., ‘Predicting information credibility in time-sensitive social media’, *Internet Res.*, vol. 23, no. 5, pp. 560–588, 2013 [Online]. Available: 10.1108/IntR-05-2012-0095.

[105] V. L. Rubin and T. Lukoianova, ‘Truth and Deception at the Rhetorical Structure Level’, *J. Assoc. Inf. Sci. Technol.*, vol. 66, no. 5, pp. 905–917, 2015 [Online]. Available: 10.1002/asi.23216.

[106] S. Kwon et al., ‘Prominent Features of Rumor Propagation in Online Social Media’, in *Proceedings - IEEE 13th International Conference on Data Mining, ICDM*, 2013, pp. 1103–1108 [Online]. Available: 10.1109/ICDM.2013.61.

[107] V. Qazvinian et al., ‘Rumor has it Identifying Misinformation in Microblogs’, in *Proceedings ofthe 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, no. July 27-31, pp. 1589–1599[Online]. Availablehttps://www.aclweb.org/anthology/D11-1147.pdf.

[108] J. Ito et al., ‘Assessment of Tweet Credibility with LDA Features’, in *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 953–958[Online]. Availablehttp://dx.doi.org/10.1145/2740908.2742569.

[109] I. E. [Ed] Chiluwa and S. A. [Ed] Samoilenko, *Handbook of Research on Deception, Fake News, and Misinformation Online*. 2019[Online]. Availablehttps://doi.org/10.4018/978-1-5225-8535-0.

[110] V. L. Rubin et al., ‘Deception Detection for News: Three Types of Fakes’, in *Proceedings of the Association for Information Science and Technology*, 2015, vol. 52, no. 1, pp. 1–4[Online]. Availablehttps://doi.org/10.1002/pra2.2015.145052010083.

[111] N. J. Conroy et al., ‘Automatic deception detection: Methods for finding fake news’, in *Proceedings of the Association for Information Science and Technology-ASIST*, 2015, vol. 52, no. 1, pp. 1–4 [Online]. Available: 10.1002/pra2.2015.145052010082.

[112] F. Yang et al., ‘Satirical News Detection and Analysis using Attention Mechanism and Linguistic Features’, in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017, pp. 1979–1989 [Online]. Available: 10.18653/v1/d17-1211.

[113] S. Argamon-Engelson et al., ‘Style-based Text Categorization: What Newspaper Am I Reading?’, in *Proceedings of AAAI Workshop on Learning for Text Categorization*, 1998, pp. 1–4[Online]. Availablehttps://www.aaai.org/Library/Workshops/1998/ws98-05-001.php.

[114] K. Popat et al., ‘Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media’, in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2019, no. April, pp. 1003–1012[Online]. Availablehttps://doi.org/10.1145/3041021.3055133.

[115] A. Barrón-Cedeño et al., ‘Proppy: Organizing the News Based on Their Propagandistic Content’, *Inf. Process. Manag.*, vol. 56, no. 5, pp. 1849–1864, 2019 [Online]. Available: 10.1016/j.ipm.2019.03.005.

[116] M. Koppel et al., ‘Measuring Differentiability: Unmasking Pseudonymous Authors’, *J. Mach. Learn. Res.*, vol. 8, no. 45, pp. 1261–1276, 2007.

[117] W. Y. Wang, ‘“Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection’, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, vol. 2, pp. 422–426[Online]. Availablehttps://www.aclweb.org/anthology/P17-2067.

[118] F. Jafariakinabad et al., ‘Syntactic Recurrent Neural Network for Authorship Attribution’, *arXiv Prepr. arXiv1902.09723*, 2019.

[119] G. da San Martino et al., ‘Fine-Grained Analysis of Propaganda in News Article’, 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)*, 2019, pp. 5636–5646[Online]. Availablehttps://www.aclweb.org/anthology/D19-1565.

[120] V. W. Feng and G. Hirst, ‘Detecting Deceptive Opinions with Profile Compatibility’, in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, no. October, pp. 338–346[Online]. Availablehttp://www.aclweb.org/anthology/I13-1039.

[121] F. Pierri and S. Ceri, ‘False News On Social Media: A Data-Driven Survey’, *ACM SIGMOD Rec.*, vol. 48, no. 2, pp. 18–32, 2019 [Online]. Available: 10.1145/3377330.3377334.

[122] S. Volkova et al., ‘Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter’, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2017, vol.

2, pp. 647–653[Online]. Availablehttps://doi.org/10.18653/v1/P17-2102.

[123] J. K. Burgoon et al., ‘Detecting Deception through Linguistic Analysis’, in *Proceedings of the 1st NSF/NIJ conference on Intelligence and security informatics*, 2003, no. June, pp. 91–101 [Online]. Available: 10.1007/3-540-44853-5_7.

[124] M. L. Newman et al., ‘Lying Words: Predicting Deception from Linguistic Styles’, *Personal. Soc. Psychol. Bull.*, vol. 29, no. 5, pp. 665–675, 2003 [Online]. Available: 10.1177/0146167203029005010.

[125] L. Zhou et al., ‘Automating Linguistics-Based Cues for Detecting Deception in Text-Based Asynchronous Computer-Mediated Communications’, *Gr. Decis. Negot. 13.1,* pp. 81–106, 2004[Online]. Availablehttps://doi.org/10.1023/B:GRUP.0000011944.62889.6f.

[126] K. Popat et al., ‘Credibility Assessment of Textual Claims on the Web’, in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management,* 2016, vol. October, pp. 2173–2178 [Online]. Available: 10.1145/2983323.2983661.

[127] B. D. Horne et al., ‘Sampling the News Producers: A Large News and Feature Data Set for the Study of the Complex Media Landscape’, in *Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018)*, 2018, vol. 12, no. 1, pp. 518–527[Online]. Availablehttps://ojs.aaai.org/index.php/ICWSM/article/view/14982.

[128] J. W. Pennebaker et al., ‘Linguistic Inquiry and Word Count (LIWC)’, *Mahw. Lawrence Erlbaum Assoc.*, vol. 71, pp. 1–24, 2001 [Online]. Available: 10.4018/978-1-60960-741-8.ch012.

[129] T. Wilson et al., ‘Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis’, in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP),* 2005, pp. 347–354 [Online]. Available: https://doi.org/10.3115/1220575.1220619.

[130] H. Treadwell, ‘The International Encyclopedia of Language and Social Interaction’, *Ref. Rev.*, vol. 30, no. 6, pp. 14–15, 2016[Online]. Availablehttps://doi.org/10.1108/RR-03-2016-0083.

[131] J. B. Hooper., ‘On assertive predicates’, in *Syntax and Semantics volume 4*, Brill, 1975, pp. 91–124 [Online]. Available: https://doi.org/10.1163/9789004368828_005.

[132] J. Thorne and A. Vlachos, ‘Automated Fact Checking: Task Formulations, Methods and Future Directions’, in *Proceedings of the 27th International Conference on Computational Linguistics,* 2018, vol. August, pp. 3346–3359[Online]. Availablehttps://www.aclweb.org/anthology/C18-1283.

[133] T. Saikh et al., ‘A Novel Approach Towards Fake News Detection: Deep Learning Augmented with Textual Entailment Features’, in *International Conference on Applications of Natural Language to Information Systems*, 2019, vol. 11608 LNCS, pp. 345–358[Online]. Availablehttp://dx.doi.org/10.1007/978-3-030-23281-8_30.

[134] D. Kang et al., ‘AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples’, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, vol. 1, no. July-Long Papers, pp. 2418–2428[Online]. Availablehttps://www.aclweb.org/anthology/P18-1225.

[135] S. R. Bowman et al., ‘A large Annotated Corpus for Learning Natural Language Inference’, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 2015, pp. 632–642[Online].

Availablehttps://www.aclweb.org/anthology/D15-1075.

[136] A. Williams et al., ‘A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference’, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers)*, 2018, vol. 1, pp. 1112–1122[Online]. Availablehttps://www.aclweb.org/anthology/N18-1101.

[137] M. Glockner et al., ‘Breaking NLI systems with sentences that require simple lexical inferences’, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2018, vol. 2, no. 3, pp. 650–655[Online]. Availablehttps://www.aclweb.org/anthology/P18-2103.

[138] F. Almeida and G. Xexéo, ‘Word Embeddings: A Survey’, *arXiv Prepr. arXiv1901.09069*, no. 1991, pp. 1–10, 2019[Online]. Availablehttp://arxiv.org/abs/1901.09069.

[139] S. Wang et al., ‘A survey of word embeddings based on deep learning’, *Computing*, vol. 102, no. 3, pp. 717–740, 2020[Online]. Availablehttps://doi.org/10.1007/s00607-019-00768-7.

[140] T. Mikolov et al., ‘Distributed Representations of Words and Phrases and their Compositionality’, in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2013, vol. 2, pp. 3111–3119[Online]. Availablehttps://dl.acm.org/doi/10.5555/2999792.2999959.

[141] L. Li et al., ‘A Rumor Events Detection Method Based on Deep Bidirectional GRU Neural Network’, *2018 3rd IEEE Int. Conf. Image, Vis. Comput. ICIVC 2018*, pp. 755–759, 2018 [Online]. Available: 10.1109/ICIVC.2018.8492819.

[142] M. Gardner et al., ‘AllenNLP: A Deep Semantic Natural Language Processing Platform’, in *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 2018, no. JULY, pp. 1–6[Online]. Availablehttps://www.aclweb.org/anthology/W18-2501.

[143] Z. Yi et al., ‘Drug-Drug Interaction Extraction via Recurrent Neural Network with Multiple Attention Layers’, in *International Conference on Advanced Data Mining and Applications, ADMA 2017*, 2017, vol. 10604, pp. 554–566 [Online]. Available: 10.1007/978-3-319-69179-4_39.

[144] X. Rong, ‘word2vec Parameter Learning Explained’, *eprint arXiv:1411.2738*, pp. 1–21, 2014[Online]. Availablehttp://arxiv.org/abs/1411.2738.

[145] T. Mikolov et al., ‘Efficient Estimation of Word Representations in Vector Space’, *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, no. October, pp. 1–12, 2013[Online]. Availablehttps://arxiv.org/abs/1301.3781.

[146] Jeffrey Pennington et al., ‘GloVe: Global Vectors for Word Representation’, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, no. October, pp. 1532–1543[Online]. Availablehttps://www.aclweb.org/anthology/D14-1162.pdf.

[147] M. Peters et al., ‘Deep Contextualized Word Representations’, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers)*, 2018, vol. 1, pp. 2227–2237[Online]. Availablehttps://www.aclweb.org/anthology/N18-1202.

[148] R. Oshikawa et al., ‘A Survey on Natural Language Processing for Fake News Detection’, in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 6086–6093[Online]. Availablehttps://www.aclweb.org/anthology/2020.lrec-1.747.

[149] D. Pisarevskaya, ‘Deception Detection in News Reports in the Russian Language: Lexics and Discourse’, in *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, 2018, pp. 74–79[Online]. Availablehttps://www.aclweb.org/anthology/W17-4213.

[150] W. C. Mann and S. A. Thompson, ‘Rhetorical Structure Theory: Toward a functional theory of text organization’, *Text & Talk*, vol. 8, no. 3. pp. 243–281, 1988 [Online]. Available: 10.1515/text.1.1988.8.3.243.

[151] M. Iruskieta et al., ‘The Annotation of the Central Unit in Rhetorical Structure Trees: A Key Step in Annotating Rhetorical Relations’, in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 466–475[Online]. Availablehttps://www.aclweb.org/anthology/C14-1045.

[152] K. Shu et al., ‘Exploiting Tri-Relationship for Fake News Detection’, *arXiv Prepr. arXiv1712.07709*, no. December, pp. 1–11, 2017 [Online]. Available: 10.1145/3289600.3290994.

[153] R. Barzilay and M. Elhadad, ‘Using Lexical Chains for Text Summarization’, in *ACL Anthology- Intelligent Scalable Text Summarization*, 1997, pp. 10–17[Online]. Availablehttps://www.aclweb.org/anthology/W97-0703.

[154] F. T. Al-Khawaldeh and V. W. Samawi, ‘Lexical Cohesion and Entailment based Segmentation for Arabic Text Summarization (LCEAS)’, *World Comput. Sci. Inf. Technol. J.*, vol. 5, no. 3, pp. 51–60, 2015[Online]. Availablehttp://oaji.net/articles/2015/567-1425407917.pdf.

[155] S. E. Toulmin, *The uses of argument (Updated edition, first published in 1958)*. Cambridge University Press2003 ,[Online]. Availablehttps://books.google.co.uk/books?hl=ar&lr=&id=8UYgegaB1S0C&oi=fnd&pg=PR7&dq=%5B60%5D%09Toulmin,+Stephen+E.+The+uses+of+argument.+Cambridge+university+press,+2003&ots=Xf_1qlDRxY&sig=u9yQZQjxniy3Cv-5m_kZeY_1K24&redir_esc=y#v=onepage&q&f=false.

[156] J. Lawrence and C. Reed, ‘Argument mining: A survey’, *Comput. Linguist.*, vol. 45, no. 4, pp. 765–818, 2019 [Online]. Available: 10.1162/COLIa00364.

[157] I. Habernal and I. Gurevych, ‘Argumentation Mining in User-Generated Web Discourse’, *Comput. Linguist.*, vol. 43, no. 1, pp. 125–179, 2017[Online]. Availablehttps://doi.org/10.1162/COLI_a_00276.

[158] K. S. Hasan and V. Ng, ‘Why are you taking this stance? Identifying and classifying reasons in ideological debates’, in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 751–762 [Online]. Available: 10.3115/v1/d14-1083.

[159] P. Reisert et al., ‘A Computational Approach for Generating Toulmin Model Argumentation’, in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015, vol. June, pp. 45–55 [Online]. Available: 10.3115/v1/w15-0507.

[160] V. D. O. Gabriel et al., ‘Argumentation-based reasoning in BDI agents using Toulmin’s model’, in *Proceedings of the Brazilian Conference on Intelligent Systems, BRACIS 2018*, 2018, pp. 378–383 [Online]. Available: 10.1109/BRACIS.2018.00072.

[161] V. de O. Gabriel et al., ‘Reasoning in BDI agents using Toulmin’s argumentation model’, *Theor. Comput. Sci.*, vol. 805, no. January, pp. 76–91, 2020[Online]. Availablehttps://doi.org/10.1016/j.tcs.2019.10.026.

[162] P. Rajendran et al., ‘Is something better than nothing? automatically predicting stance-based arguments using deep learning and small labelled dataset’, in

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, vol. 2, pp. 28–34 [Online]. Available: 10.18653/v1/n18-2005.

[163] P. Rajendran et al., 'Contextual Stance Classification of Opinions: A Step towards Enthymeme Reconstruction in Online Reviews', in *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, 2016, no. August, pp. 31–39 [Online]. Available: 10.18653/v1/w16-2804.

[164] K. Singh et al., 'Improving Evidence Detection by Leveraging Warrants', in *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, 2019, no. November, pp. 57–62 [Online]. Available: 10.18653/v1/d19-6610.

[165] U. Khurana, 'The Linguistic Features of Fake News Headlines and Statements', University of Amsterdam, 2017.

[166] N. Hassan et al., 'Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster', in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2017, pp. 1803–1812 [Online]. Available: 10.1145/3097983.3098131.

[167] S. Das Bhattacharjee et al., 'Active Learning Based News Veracity Detection with Feature Weighting and Deep-shallow Fusion', in *Proceedings of the IEEE International Conference on Big Data (BIGDATA),* 2017, vol. 1, pp. 556–565 [Online]. Available: 10.1109/BigData.2017.8257971.

[168] Y. Liu and Y. F. B. 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*, 2018, vol. 32, no. 1, pp. 354–361[Online]. Availablehttps://ojs.aaai.org/index.php/AAAI/article/view/11268.

[169] Brian Xu, 'Combating Fake News with Adversarial Domain Adaptation and Neural Models', MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2019[Online]. Availablehttps://hdl.handle.net/1721.1/121689.

[170] D. Fallis, 'What Is Disinformation?', in *LIBRARY TRENDS*, 2015, vol. 63, no. 3, pp. 401–426 [Online]. Available: 10.1353/lib.2015.0014.

[171] F. Rosenblatt, 'The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain', *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958[Online]. Availablehttps://doi.org/10.1037/h0042519.

[172] K. Dey et al., 'Topical Stance Detection for Twitter: A Two-Phase LSTM Model Using Attention', *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),* vol. 10772, pp. 529–536, 2018 [Online]. Available: 10.1007/978-3-319-76941-7_40.

[173] Y. Kim, 'Convolutional Neural Networks for Sentence Classification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 2014, pp. 1746–1751[Online]. Availablehttps://www.aclweb.org/anthology/D14-1181.

[174] T. Niven and H. Y. Kao, 'Probing Neural Network Comprehension of Natural Language Arguments', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* 2019, pp. 4658–4664[Online]. Availablehttps://www.aclweb.org/anthology/P19-1459.

[175] S. Li et al., 'Initializing Convolutional Filters with Semantic Features for Text Classification', in *Proceedings ofthe 2017 Conference on Empirical Methods in*

Natural Language Processing EMNLP, 2017, pp. 1884–1889[Online]. Availablehttps://www.aclweb.org/anthology/D17-1201.

[176] Z. Qian et al., 'Speculation and Negation Scope Detection via Convolutional Neural Networks', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 815–825[Online]. Availablehttps://www.aclweb.org/anthology/D16-1078.

[177] D. K. Duvenaud et al., 'Convolutional Networks on Graphs for Learning Molecular Fingerprints', in *Proceedings of Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015, vol. 2, no. December, pp. 2224–2232.

[178] Michaël Defferrard et al., 'Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering', *Adv. Neural Inf. Process. Syst. 29 (NIPS 2016)*, pp. 3844–3852, 2016.

[179] T. N. Kipf and M. Welling, 'Semi-Supervised Classification with Graph Convolutional Networks', in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017, pp. 1–14[Online]. Availablehttps://arxiv.org/abs/1609.02907.

[180] J. Bruna et al., 'Spectral Networks and Deep Locally Connected Networks on Graphs', in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014, pp. 1–14[Online]. Availablehttp://arxiv.org/abs/1312.6203.

[181] G. Hu et al., *Multi-depth Graph Convolutional Networks for Fake News Detection*, vol. 11838. Springer, Cham, 2019[Online]. Availablehttp://dx.doi.org/10.1007/978-3-030-32233-5_54.

[182] A. Graves et al., 'Speech Recognition with Deep Recurrent Neural Networks', in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, 2013, no. 3, pp. 6645–6649 [Online]. Available: 10.1109/ICASSP.2013.6638947.

[183] Y. Zhang et al., 'Hierarchical Attention Transfer Network for Cross-domain Sentiment Classification', in *Proceeding of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2019, pp. 5852–5859[Online]. Availablehttp://hdl.handle.net/1783.1/90256.

[184] S. Gao et al., 'Hierarchical Convolutional Attention Networks for Text Classification', in *Proceedings ofthe 3rd Workshop on Representation Learning for NLP*, 2018, pp. 11–23[Online]. Availablehttps://www.aclweb.org/anthology/W18-3002.

[185] G. E. Hinton and R. R. Salakhutdinov, 'Reducing the Dimensionality of Data with Neural Networks', *Science (80-.).*, vol. 313, no. 5786, pp. 504–507, 2006 [Online]. Available: 10.1126/science.1127647.

[186] D. P. Kingma and M. Welling, 'Stochastic Gradient Variational Bayes and the Variational Autoencoder', *Second Int. Conf. Learn. Represent. ICLR*, vol. 19, pp. 1–31, 2014[Online]. Availablehttps://www.semanticscholar.org/paper/Stochastic-Gradient-VB-and-the-Variational-Kingma-Welling/eaa6bf5334bc647153518d0205dca2f73aea971e.

[187] Z. Hu et al., 'Toward Controlled Generation of Text', in *34th International Conference on Machine Learning, ICML 2017*, 2017, vol. 4, no. PMLR 70, pp. 2503–2513.

[188] D. P. Kingma and M. Welling, 'Auto-Encoding Variational Bayes', *Neuroimage*, vol. 147, no. Ml, pp. 302–313, 2013[Online]. Availablearxiv:1312.6114.

[189] D. P. Kingma et al., ‘Semi-supervised learning with deep generative models’, *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3581–3589, 2014[Online]. Availablehttp://arxiv.org/abs/1406.5298%0D.

[190] I. V. Serban et al., ‘A hierarchical latent variable encoder-decoder model for generating dialogues’, in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017, pp. 3295–3301.

[191] T. Zhao et al., ‘Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders’, in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, vol. 1, pp. 654–664 [Online]. Available: 10.18653/v1/P17-1061.

[192] X. Shen et al., ‘A Conditional Variational Framework for Dialog Generation’, in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, vol. 2, pp. 504–509 [Online]. Available: 10.18653/v1/P17-2080.

[193] X. Zhou and W. Y. Wang, ‘Mojitalk: Generating emotional responses at scale’, in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, vol. 1, pp. 1128–1137 [Online]. Available: 10.18653/v1/p18-1104.

[194] D. Moyer et al., ‘Invariant Representations without Adversarial Training’, *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 9084–9093, 2018.

[195] S. M. Padnekar et al., ‘BiLSTM-Autoencoder Architecture for Stance Prediction’, in *2020 International Conference on Data Science and Engineering, ICDSE 2020*, 2020, pp. 1–5 [Online]. Available: 10.1109/ICDSE50459.2020.9310133.

[196] B. Gong et al., ‘Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation’, in *Proceedings of the 30th International Conference on Machine Learning*, 2013, vol. 28, pp. 222–230[Online]. Availablehttp://www.cs.utexas.edu/~grauman/papers/landmark.pdf.

[197] S. J. Pan et al., ‘Domain Adaptation via Transfer Component Analysis’, *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011 [Online]. Available: 10.1109/TNN.2010.2091281.

[198] A. Vani, ‘Adversarial Discrete Sequence Generation’, 2017[Online]. Availablehttps://ankitvani.com/reports/fergus-report.pdf.

[199] Y. Tuan and H. Lee, ‘Improving Conditional Sequence Generative Adversarial Networks by Stepwise Evaluation’, *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 4, pp. 788–798, 2019.

[200] K. Zhang et al., ‘Domain Adaptation under Target and Conditional Shift’, *Proc. 30th Int. Conf. Mach. Learn. 2013*, vol. 28, no. PART 3, pp. 819–827, 2013.

[201] N. Courty et al., ‘Joint Distribution Optimal Transportation for Domain Adaptation’, in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 3730–3739[Online]. Availablehttp://onlinelibrary.wiley.com/doi/10.1002/jhm.920/pdf%0Ahttp://ovidsp.ovid.com/ovidweb.cgi?T=JS&PAGE=reference&D=emed12&NEWS=N&AN=70423585.

[202] L. Yu et al., ‘SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient’, *Proc. Thirty-First AAAI Conf. Artif. Intell.*, vol. 31, no. 1, pp. 2852–2858, 2017[Online]. Availablehttps://ojs.aaai.org/index.php/AAAI/article/view/10804.

[203] K. Lin et al., ‘Adversarial Ranking for Language Generation’, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017,

pp. 3158–3168.

[204] M. J. Kusner and J. M. Hernández-Lobato, ‘GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution’, *CoRR*, vol. abs/1611.0, 2016[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1611.html#KusnerH16.

[205] L. Chen et al., ‘Adversarial Text Generation via Feature-Mover’s Distance’, in *Advances in Neural Information Processing Systems -Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018, vol. 2018-Decem, no. NeurIPS, pp. 4671–4682.

[206] H. Azarbonyad et al., ‘Domain Adaptation for Commitment Detection in Email’, in *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 2019, no. January, pp. 672–680[Online]. Availablehttps://doi.org/10.1145/3289600.3290984.

[207] A. Rios et al., ‘Generalizing Biomedical Relation Classification with Neural Adversaria Domain Adaptation’, *Bioinformatics*, vol. 34, no. 17, pp. 2973–2981, 2018 [Online]. Available: 10.1093/bioinformatics/bty190.

[208] D. Hu, ‘An Introductory Survey on Attention Mechanisms in NLP Problems’, in *Proceedings of SAI Intelligent Systems Conference IntelliSys 2019: Intelligent Systems and Applications- Advances in Intelligent Systems and Computing*, 2020, vol. 1038, pp. 432–448[Online]. Availablehttps://doi.org/10.1007/978-3-030-29513-4_31.

[209] F. Yu et al., ‘Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts’, *Comput. Secur.*, vol. 83, pp. 106–121, 2019[Online]. Availablehttps://doi.org/10.1016/j.cose.2019.02.003.

[210] Y. Ren and J. Zhang, ‘Fake News Detection on News-Oriented Heterogeneous Information Networks through Hierarchical Graph Attention’, *arXiv Prepr. arXiv2002.04397*, pp. 1–9, 2021[Online]. Availablehttps://ui.adsabs.harvard.edu/abs/2020arXiv200204397R/abstract.

[211] P. Zhou et al., ‘Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification’, in *Proceedings ofthe 54th Annual Meeting ofthe Association for Computational Linguistics (Short papers)*, 2016, vol. 2, pp. 207–212 [Online]. Available: 10.18653/v1/p16-2034.

[212] A. Vaswani et al., ‘Attention Is All You Need’, in *Proceeding of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 5998–6008.

[213] G. Hinton et al., ‘Matrix Capsules with EM Routing’, in *Proceeding to the 6th International conference on learning representations - ICLR*, 2018, pp. 1–15 [Online]. Available: 10.2514/1.562.

[214] S. Sabour et al., ‘Dynamic Routing Between Capsules’, in *31st Conference on Neural Information Processing Systems (NIPS 2017)- Advances in neural information processing systems*, 2017, pp. 3857–3867[Online]. Availablehttps://papers.nips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf.

[215] D. K. Jain et al., ‘Deep Refinement: capsule network with attention mechanism-based system for text classification’, *Neural Comput. Appl.*, vol. 32, pp. 1839–1856, 2020 [Online]. Available: https://doi.org/10.1007/s00521-019-04620-z.

[216] S. Li et al., ‘Capsules Based Chinese Word Segmentation for Ancient Chinese Medical Books’, *IEEE Access*, vol. 6, pp. 70874–70883, 2018 [Online]. Available: 10.1109/ACCESS.2018.2881280.

[217] Y. Wu et al., ‘Siamese Capsule Networks with Global and Local Features for Text Classification’, *Neurocomputing*, vol. 390, pp. 88–98, 2020[Online]. Availablehttps://doi.org/10.1016/j.neucom.2020.01.064.

[218] Y. Du et al., ‘A Novel Capsule Based Hybrid Neural Network for Sentiment Classification’, *IEEE Access*, vol. 7, pp. 39321–39328, 2019 [Online]. Available: 10.1109/ACCESS.2019.2906398.

[219] J. Kim et al., ‘Text Classification using Capsules’, *Neurocomputing*, vol. 376, pp. 214–221, Feb. 2020 [Online]. Available: 10.1016/j.neucom.2019.10.033.

[220] H. Yin et al., ‘Capsule Network with Identifying Transferable Knowledge for Cross-Domain Sentiment Classification’, *IEEE Access*, vol. 7, pp. 153171–153182, 2019 [Online]. Available: 10.1109/ACCESS.2019.2948628.

[221] M. Yang et al., ‘Investigating the Transferring Capability of Capsule Networks for Text Classification’, *Neural Networks*, vol. 118, pp. 247–261, 2019[Online]. Availablehttps://doi.org/10.1016/j.neunet.2019.06.014.

[222] A. Kumar et al., ‘Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM’, *IEEE Access*, vol. 8, pp. 6388–6397, 2020 [Online]. Available: 10.1109/ACCESS.2019.2963630.

[223] G.-A. Vlad et al., ‘Sentence-Level Propaganda Detection in News Articles with Transfer Learning and BERT-BiLSTM-Capsule Model’, in *Proceedings ofthe 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019, pp. 148–154[Online]. Availablehttps://www.aclweb.org/anthology/D19-5022.

[224] Y. Wang et al., ‘Sentiment Analysis by Capsules’, in *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*, 2018, vol. 2, pp. 1165–1174 [Online]. Available: 10.1145/3178876.3186015.

[225] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second edi. MIT press, 2018[Online]. Availablehttps://books.google.co.uk/books?hl=ar&lr=&id=uWV0DwAAQBAJ&oi=fnd&pg=PR7&dq=%5B33%5D%09Sutton,+Richard+S.,+and+Andrew+G.+Barto.+Reinforcement+learning:+An+introduction.+MIT+press,+2018&ots=minJq6-Yg8&sig=l-l3EnSsFl9aYo_yOyXLJG3eKN0&redir_esc=y#v=onepage&q.

[226] J. Li et al., ‘Deep Reinforcement Learning for Dialogue Generation’, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing-EMNLP 2016*, 2016, no. 4, pp. 1192–1202[Online]. Availablehttps://www.aclweb.org/anthology/D16-1127.

[227] T. Zhang et al., ‘Learning Structured Representation for Text Classification via Reinforcement Learning’, in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 6053–6060.

[228] A. S. Vezhnevets et al., ‘FeUdal Networks for Hierarchical Reinforcement Learning’, in *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 7, pp. 3540–3549.

[229] M.-Y. Liu and O. Tuzel, ‘Coupled Generative Adversarial Networks’, in *30th Conference on Neural Information Processing Systems (NIPS 2016),* 2016, pp. 469–477[Online]. Availablehttps://papers.nips.cc/paper/2016/file/502e4a16930e414107ee22b6198c578f-Paper.pdf.

[230] K. Bousmalis et al., ‘Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2019, pp. 3722–3731.

[231] S. Sankaranarayanan et al., ‘Generate to Adapt: Aligning Domains Using Generative Adversarial Networks’, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* 2018, pp. 8503–8512 [Online]. Available: 10.1109/CVPR.2018.00887.

[232] P. Russo et al., ‘From Source to Target and Back: Symmetric Bi-Directional Adaptive GAN’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2018, pp. 8099–8108 [Online]. Available: 10.1109/CVPR.2018.00845.

[233] Y. Ganin et al., ‘Domain-Adversarial Training of Neural Networks’, in *The Journal of Machine Learning ResearchVol Domain-adversarial training of neural networks,* 2017, vol. 17, no. 1, pp. 1–35[Online]. Availablehttps://doi.org/10.1007/978-3-319-58347-1_10.

[234] E. Tzeng et al., ‘Adversarial Discriminative Domain Adaptation’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017, pp. 7167–7176 [Online]. Available: 10.1109/CVPR.2017.316.

[235] X. Chen et al., ‘Adversarial Deep Averaging Networks for Cross-Lingual Sentiment Classification’, *Trans. Assoc. Comput. Linguist.,* vol. 6, pp. 557–570, 2018 [Online]. Available: 10.1162/tacl_a_00039.

[236] T. Gui et al., ‘Part-of-Speech Tagging for Twitter with Adversarial Neural Networks’, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* 2017, pp. 2411–2420[Online]. Availablehttps://www.aclweb.org/anthology/D17-1256.

[237] Y. Zhang et al., ‘Aspect-augmented Adversarial Networks for Domain Adaptation’, *Trans. Assoc. Comput. Linguist.,* vol. 5, pp. 515–528, 2017 [Online]. Available: 10.1162/tacl_a_00077.

[238] L. Fu et al., ‘Domain Adaptation for Relation Extraction with Domain Adversarial Neural Network’, in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Short Papers)*, 2017, vol. 2, pp. 425–429[Online]. Availablehttps://www.aclweb.org/anthology/I17-2072.

[239] S. Joty et al., ‘Cross-language Learning with Adversarial Neural Networks: Application to Community Question Answering’, in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017),* 2017, pp. 226–237[Online]. Availablehttps://www.aclweb.org/anthology/K17-1024.

[240] R. Xu and Y. Yang, ‘Cross-lingual Distillation for Text Classification’, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers),* 2017, vol. 1, pp. 1415–1425[Online]. Availablehttps://www.aclweb.org/anthology/P17-1130.

[241] M. Long et al., ‘Deep Transfer Learning with Joint Adaptation Networks’, in *Proceedings of the 34th International Conference on Machine Learning,* 2017, vol. 70, pp. 2208–2217.

[242] Y. Cao et al., ‘Unsupervised Domain Adaptation With Distribution Matching Machines’, in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence,* 2018, vol. 32, no. 1, pp. 2795–2802[Online]. Availablehttps://ojs.aaai.org/index.php/AAAI/article/view/11792.

[243] K. Sohn et al., ‘Unsupervised Domain Adaptation for Distance Metric Learning’, in *7th International Conference on Learning Representations, ICLR 2019,* 2019[Online]. Availablehttps://ojs.aaai.org/index.php/AAAI/article/view/11792.

[244] K. Bousmalis et al., ‘Domain Separation Networks’, in *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS 2016),*

2016, pp. 343–351[Online]. Availablehttps://dl.acm.org/doi/10.5555/3157096.3157135.

[245] X. Chen and C. Cardie, 'Multinomial Adversarial Networks for Multi-Domain Text Classification', in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers)*, 2018, vol. 1, pp. 1226–1240[Online]. Availablehttps://www.aclweb.org/anthology/N18-1111.

[246] Y. Ganin and V. Lempitsky, 'Unsupervised Domain Adaptation by Backpropagation', in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015, vol. 37, pp. 1180–1189[Online]. Availablehttps://dl.acm.org/doi/10.5555/3045118.3045244.

[247] P. Liu et al., 'Adversarial Multi-task Learning for Text Classification', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2017, vol. 1, pp. 1–10[Online]. Availablehttps://www.aclweb.org/anthology/P17-1001.

[248] D. Wang et al., 'On Truth Discovery in Social Sensing: A Maximum Likelihood Estimation Approach', in *IPSN'12 - Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, 2012, no. April, pp. 233–244[Online]. Availablehttps://doi.org/10.1145/2185677.2185737.

[249] A. Zubiaga et al., 'Detection and Resolution of Rumours in Social Media: A Survey', *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–36, 2018[Online]. Availablehttps://doi.org/10.1145/3161603.

[250] A. Zubiaga et al., 'Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads', *PLoS ONE*, vol. 11, no. 3. pp. 1–29, 2016 [Online]. Available: 10.1371/journal.pone.0150989.

[251] K. Sharma et al., 'Combating Fake News: A Survey on Identification and Mitigation Techniques', *ACM Trans. Intell. Syst. Technol.*, vol. 37, no. 4, pp. 1–41, 2019[Online]. Availablehttps://doi.org/10.1145/3305260.

[252] K. Shu et al., 'Fake News Detection on Social Media: A Data Mining Perspective', *ACM SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, 2017 [Online]. Available: 10.1145/3137597.3137600.

[253] X. Liu et al., 'Real-time Rumor Debunking on Twitter', in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, no. October, pp. 1867–1870[Online]. Availablehttps://doi.org/10.1145/2806416.2806651.

[254] J. 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*, 2015, vol. 19-23-Oct-, no. Cikm, pp. 1751–1754[Online]. Availablehttps://doi.org/10.1145/2806416.2806607.

[255] Z. Zhao et al., 'Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts', in *Proceedings of the 24th i International World Wide Web Conference*, 2015, pp. 1395–1405[Online]. Availablehttp://dx.doi.org/10.1145/2736277.2741637.

[256] A. Zubiaga et al., (2016,), *PHEME Dataset of Rumours and Non-rumours*, figshare. Dataset. [Online]. Available: https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619/1. [Accessed: 20 Nov. 2020].

[257] W. Ferreira and A. Vlachos, 'Emergent: A Novel Data-set for Stance

Classification', in *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016, vol. June 12-17, no. 1, pp. 1163–1168 [Online]. Available: 10.18653/v1/n16-1138.

[258] A. Rajadesingan and H. Liu, 'Identifying Users with Opposing Opinions in Twitter Debates', in *International conference on social computing, behavioral-cultural modeling, and prediction,* 2014, vol. 8393 LNCS, pp. 153–160[Online]. Availablehttps://doi.org/10.1007/978-3-319-05579-4_19.

[259] L. Wang and C. Cardie, 'Improving Agreement and Disagreement Identification in Online Discussions with A Socially-Tuned Sentiment Lexicon', in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis,* 2014, pp. 97–106 [Online]. Available: 10.3115/v1/W14-2617.

[260] R. Baly et al., 'Integrating Stance Detection and Fact Checking in a Unified Corpus', in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,(Short Papers),* 2018, vol. 2, pp. 21–27[Online]. Availablehttps://www.aclweb.org/anthology/N18-2004.

[261] Fakenewschallenge.org, *FAKE NEWS CHALLENGE STAGE 1 (FNC-I): STANCE DETECTION*, FNC. [Online]. Available: http://www.fakenewschallenge.org/. [Accessed: 30 Nov. 2020].

[262] M. Nadeem et al., 'FAKTA: An Automatic End-to-End Fact Checking System', in *Proceedings of NAACL HLT 2019 - Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - of the Demonstrations Session,* 2019, pp. 78–83.

[263] D. Küçük and C. Fazli, 'Stance detection: A survey', *ACM Comput. Surv.,* vol. 53, no. 1, pp. 1–37, 2020 [Online]. Available: 10.1145/3369026.

[264] S. Dungs et al., 'Can Rumour Stance Alone Predict Veracity?', in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3360–3370[Online]. Availablehttps://aclweb.org/anthology/papers/C/C18/C18-1284/.

[265] D. Sridhar et al., 'Collective Stance Classification of Posts in Online Debate Forums', in *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, 2015, no. June, pp. 109–117 [Online]. Available: 10.3115/v1/w14-2715.

[266] J. Ebrahimi et al., 'A Joint Sentiment-Target-Stance Model for Stance Classification in Tweets', in *Proceedings of COLING 2016- 26th International Conference on Computational Linguistics, Technical Papers*, 2016, no. December, pp. 2656–2665.

[267] I. Augenstein et al., 'Stance Detection with Bidirectional Conditional Encoding', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* 2016, pp. 876–885 [Online]. Available: 10.18653/v1/d16-1084.

[268] J. Du et al., 'Stance Classification with Target-Specific Neural Attention Networks', in *26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, 2017, pp. 3988–3994[Online]. Availablehttps://doi.org/10.24963/ijcai.2017/557.

[269] J. Devlin et al., 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in *NAACL -HLT 2019 - 2019 Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 2019, vol. 1, no. Mlm, pp. 4171–4186 [Online]. Available: 10.18653/v1/N19-1423.

[270] B. Xu et al., 'Adversarial Domain Adaptation for Stance Detection', in *32nd*

Conference on Neural Information Processing Systems (NIPS 2018), 2019, no. Nips, pp. 1–6.

[271] A. Hanselowski et al., (2017,), *Description of the System Developed by Team Athene in the FNC-1*, Technical report. [Online]. Available: https://github.com/hanselowski/athene_system/blob/master/system_description_athene.pdf. [Accessed: 24 Nov. 2020].

[272] Q. Zhang et al., ‘From Stances’ Imbalance to Their Hierarchical Representation and Detection’, in *Proceedings of the World Wide Web Conference, WWW 2019*, 2019, vol. May 13-17, no. 1, pp. 2323–2332 [Online]. Available: 10.1145/3308558.3313724.

[273] S. Bajaj, ‘“ The Pope Has a New Baby !” Fake News Detection Using Deep Learning’, *CS 224N*, pp. 1–8, 2017[Online]. Availablehttps://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2710385.pdf.

[274] Q. Zhang et al., ‘Ranking-based Method for News Stance Detection’, in *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, 2018, pp. 41–42 [Online]. Available: 10.1145/3184558.3186919.

[275] R. Saurí and J. Pustejovsky, ‘FactBank: A Corpus Annotated with Event Factuality’, *Lang. Resour. Eval.*, vol. 43, no. 3, pp. 227–268, 2009 [Online]. Available: 10.1007/s10579-009-9089-9.

[276] P. Bourgonje et al., ‘From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles’, in *Proceedings ofthe 2017 EMNLP Workshop on Natural Language Processing meets Journalism*, 2017, pp. 84–89 [Online]. Available: 10.18653/v1/w17-4215.

[277] X. Wang et al., ‘Relevant Document Discovery for Fact-Checking Articles’, in *Companion Proceedings of the The Web Conference 2018, International World Wide Web Conferences Steering Committee*, 2018, pp. 525–533[Online]. Availablehttps://doi.org/10.1145/3184558.3188723.

[278] J. Li et al., ‘Unsupervised Streaming Feature Selection in Social Media’, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, no. October 13, pp. 1041–1050[Online]. Availablehttp://dx.doi.org/10.1145/2806416.2806501.

[279] P. Wei et al., ‘Modeling Conversation Structure and Temporal Dynamics for Jointly Predicting Rumor Stance and Veracity’, in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 4787–4798 [Online]. Available: 10.18653/v1/d19-1485.

[280] M. Glenski et al., ‘Identifying and Understanding User Reactions to Deceptive and Trusted Social News Sources’, *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*, vol. 2, no. 1, pp. 176–181, 2018 [Online]. Available: 10.18653/v1/p18-2029.

[281] M. Mendoza et al., ‘Twitter Under Crisis: Can we trust what we RT?’, in *SOMA 2010 - Proceedings of the 1st Workshop on Social Media Analytics*, 2010, pp. 71–79 [Online]. Available: 10.1145/1964858.1964869.

[282] R. Procter et al., ‘Reading the Riots on Twitter: Methodological Innovation for the Analysis of Big Data’, *Int. J. Soc. Res. Methodol. Comput. Soc. Sci. Res. Strateg. Des. Methods*, vol. 16, no. 3, pp. 197–214, 2013 [Online]. Available: 10.1080/10439463.2013.780223.

[283] Q. Zhang et al., ‘Automatic Detection of Rumor on Social Network’, *NLPCC2015*,

Nat. Lang. Process. Chinese Comput. Springer, Cham, vol. LNAI 9362, pp. 113–122, 2015 [Online]. Available: 10.1007/978-3-319-25207-0_10.

[284] A. Zubiaga et al., ‘Stance Classification in Rumours as a Sequential Task Exploiting the Tree Structure of Social Media Conversations’, in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2438–2448[Online]. Availablehttps://www.aclweb.org/anthology/C16-1230.

[285] A. Khandelwal, ‘Fine-Tune Longformer for Jointly Predicting Rumor Stance and Veracity’, *arXiv Prepr.*, 2020[Online]. Availablehttp://arxiv.org/abs/2007.07803.

[286] G. Gorrell et al., ‘RumourEval 2019: Determining Rumour Veracity and Support for Rumours’, in *Proceedings ofthe 13th International Workshop on Semantic Evaluation (SemEval-2019)*, 2019, pp. 845–854.

[287] R. Yang et al., ‘BLCU_NLP at SemEval-2019 Task 7: An Inference Chain-based GPT Model for Rumour Evaluation’, in *Proceedings ofthe 13th International Workshop on Semantic Evaluation (SemEval-2019)*, 2019, no. June 6-7, pp. 1090–1096 [Online]. Available: 10.18653/v1/s19-2191.

[288] M. Fajcik et al., ‘BUT-FIT at SemEval-2019 Task 7: Determining the Rumour Stance with Pre-Trained Deep Bidirectional Transformers’, in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. 1097–1104 [Online]. Available: 10.18653/v1/s19-2192.

[289] I. Baris et al., ‘CLEARumor at SemEval-2019 Task 7: ConvoLving ELMo against rumors’, in *3th International Workshop on Semantic Evaluation*, 2019, no. June 06-07[Online]. Availablehttp://eprints.soton.ac.uk/id/eprint/430196.

[290] A. Radford et al., (2018,), *Improving Language Understanding with Unsupervised Learning*, Technical report, Open AI. [Online]. Available: https://openai.com/blog/language-unsupervised/. [Accessed: 20 Nov. 2020].

[291] T. Mitra and E. Gilbert, ‘CREDBANK: A Large-scale Social Media Corpus With Associated Credibility Annotations’, *Proc. Ninth Int. AAAI Conf. Web Soc. Media -ICWSM*, pp. 258–267, 2015 [Online]. Available: 10.1017/s0002731600003863.

[292] M. Samadi et al., ‘ClaimEval: Integrated and Flexible Framework for Claim Evaluation Using Credibility of Sources’, in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, no. February, pp. 222–228[Online]. Availablehttp://dblp.uni-trier.de/db/conf/aaai/aaai2016.html#SamadiTVB16.

[293] P. Nakov et al., ‘Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims’, *Int. Conf. Cross-Language Eval. Forum Eur. Lang. - CLEF 2018.*, vol. vol 11018, no. December, pp. 372–387, 2018 [Online]. Available: 10.1007/978-3-319-98932-7_32.

[294] H. Karimi et al., ‘Multi-Source Multi-Class Fake News Detection’, in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1546–1557[Online]. Availablehttps://aclanthology.coli.uni-saarland.de/papers/C18-1131/c18-1131.

[295] T. Alhindi et al., ‘Where is Your Evidence: Improving Fact-checking by Justification Modeling’, in *Proceedings ofthe First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 85–90 [Online]. Available: 10.18653/v1/w18-5513.

[296] W. Yin and D. Roth, ‘Twowingos: A two-wing optimization strategy for evidential claim verification’, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020, pp. 105–114 [Online]. Available: 10.18653/v1/d18-1010.

[297] M. Seo et al., ‘Bidirectional Attention Flow for Machine Comprehension’, in *ICLR 2017 conference submission*, 2016, pp. 1–13[Online]. Availablehttp://arxiv.org/abs/1611.01603.

[298] Y. Nie et al., ‘Combining Fact Extraction and Verification with Neural Semantic Matching Networks’, in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 01, pp. 6859–6866 [Online]. Available: 10.1609/aaai.v33i01.33016859.

[299] A. Soleimani et al., ‘BERT for Evidence Retrieval and Claim Verification’, *J. M. Jose al. ECIR 2020, LNCS 12036*, no. 3, pp. 359–366, 2020[Online]. Availablehttp://dx.doi.org/10.1007/978-3-030-45442-5_45.

[300] K. Baker et al., ‘A Modality lexicon and its use in automatic tagging’, in *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*, 2010, pp. 1402–1407.

[301] K. Lee et al., ‘Event Detection and Factuality Assessment with Non-Expert Supervision’, in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, no. September, pp. 1643–1648 [Online]. Available: 10.18653/v1/d15-1189.

[302] V. Beretta et al., ‘Truth Selection for Truth Discovery Models Exploiting Ordering Relationship Among Values’, *Knowledge-Based Syst.*, vol. 159, pp. 298–308, 2018[Online]. Availablehttps://doi.org/10.1016/j.knosys.2018.06.023.

[303] S. L. Blodgett et al., ‘Demographic dialectal variation in social media: A case study of African-American English’, in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016, pp. 1119–1130 [Online]. Available: 10.18653/v1/d16-1120.

[304] G. Stanovsky et al., ‘Integrating Deep Linguistic Features in Factuality Prediction over Unified Datasets’, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2017, vol. 2, pp. 352–357[Online]. Availablehttps://doi.org/10.18653/v1/P17-2056%0D.

[305] O. Enayet and S. R. El-Beltagy, ‘NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter.’, in *Proceedings ofthe 11th International Workshop on Semantic Evaluations (SemEval-2017)*, 2017, pp. 470–474 [Online]. Available: 10.18653/v1/s17-2082.

[306] E. Kochkina et al., ‘All-in-one: Multi-task Learning for Rumour Verification’, in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3402–3413[Online]. Availablehttps://www.aclweb.org/anthology/C18-1288.

[307] J. Ma et al., ‘Detect Rumor and Stance Jointly by Neural Multi-task Learning’, in *Proceedings of the Web Conference (WWW 2018 Companion)*, 2018, pp. 585–593[Online]. Availablehttps://ink.library.smu.edu.sg/sis_research/4562.

[308] L. Poddar et al., ‘Predicting Stances in Twitter Conversations for Detecting Veracity of Rumors: a Neural Approach’, in *Proceedings of the 30th International Conference on Tools with Artificial Intelligence, ICTAI*, 2018, vol. 2018-Novem, pp. 65–72 [Online]. Available: 10.1109/ICTAI.2018.00021.

[309] M. E. Peters and A. Cohan, ‘Longformer: The Long-Document Transformer’, *arXiv Prepr. arXiv2004.05150v1*, 2020.

[310] Q. Li et al., ‘eventAI at SemEval-2019 Task 7: Rumor Detection on Social Media by Exploiting Content, User Credibility and Propagation Information’, in *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, 2019, pp. 855–859 [Online]. Available: 10.18653/v1/s19-2148.

[311] S. Zhi et al., ‘Modeling Truth Existence in Truth Discovery’, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2015, vol. August, pp. 1543–1552[Online]. Availablehttps://doi.org/10.1145/2783258.2783339.

[312] S. Lyu et al., ‘Truth Discovery by Claim and Source Embedding’, *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 8, pp. 1–12, 2019 [Online]. Available: 10.1109/tkde.2019.2936189.

[313] Y. Li et al., ‘A Survey on Truth Discovery’, *ACM SIGKDD Explor. Newsl.*, vol. 17, no. 2, pp. 1–16, 2016 [Online]. Available: 10.1145/2897350.2897352.

[314] J. Pasternack and D. Roth, ‘Knowing What to Believe (when you already know something)’, in *Proceedings ofthe 23rd International Conference on Computational Linguistics (Coling 2010)*, 2010, vol. 2, no. August, pp. 877–885.

[315] Y. Li et al., ‘On the Discovery of Evolving Truth’, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, vol. 2015-Augus, pp. 675–684 [Online]. Available: 10.1145/2783258.2783277.

[316] J. Marshall et al., ‘A Neural Network Approach for Truth Discovery in Social Sensing’, in *Proceedings - 14th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2017*, 2017, pp. 343–347 [Online]. Available: 10.1109/MASS.2017.26.

[317] X. Yin et al., ‘Truth Discovery with Multiple Conflicting Information Providers on the Web’, *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 6, pp. 796–808, 2008 [Online]. Available: 10.1109/TKDE.2007.190745.

[318] X. L. Dong et al., ‘Integrating Conflicting Data: The Role of Source Dependence’, in *Proceedings of the VLDB Endowment*, 2009, vol. 2, no. 1, pp. 550–561[Online]. Availablehttps://doi.org/10.14778/1687627.1687690.

[319] J. Pasternack and D. Roth, ‘Making Better Informed Trust Decisions with Generalized Fact-Finding’, in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence Making*, 2011, pp. 2324–2329 [Online]. Available: 10.5591/978-1-57735-516-8/IJCAI11-387.

[320] A. Galland et al., ‘Corroborating Information from Disagreeing Views’, in *Proceedings of the third ACM International Conference on Web Search and Data Mining (WSDM),* 2010, pp. 131–140[Online]. Availablehttps://doi.org/10.1145/1718487.1718504.

[321] F. Li et al., ‘Entity Profiling with Varying Source Reliabilities’, in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining,* 2014, no. August, pp. 1146–1155[Online]. Availablehttps://doi.org/10.1145/2623330.2623685.

[322] C. Dai et al., ‘An Approach to Evaluate Data Trustworthiness Based on Data Provenance’, in *Proceedings of the 5th VLDB Workshop on Secure Data Management*, 2008, pp. 82–98 [Online]. Available: 10.1007/978-3-540-85259-9_6.

[323] X. 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*, 2014, pp. 601–610[Online]. Availablehttps://doi.org/10.1145/2623330.2623623.

[324] Q. Li et al., ‘A Confidence-Aware Approach for Truth Discovery on Long-Tail Data’, *Proc. VLDB Endow.*, vol. 8, no. 4, pp. 425–436, 2014[Online]. Availablehttps://doi.org/10.14778/2735496.2735505.

[325] Q. Li et al., ‘Resolving Conflicts in Heterogeneous Data by Truth Discovery and Source Reliability Estimation’, in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD ’14)*, 2014, pp. 1187–1198[Online]. Availablehttps://doi.org/10.1145/2588555.2610509.

[326] D. Zhou et al., ‘Learning from the Wisdom of Crowds by Minimax Entropy’, in *Proceedings of the 25th International Conference on Neural Information Processing Systems NIPS*, 2012, vol. 2, no. December, pp. 2195–2203[Online]. Availablehttp://dblp.uni-trier.de/db/conf/nips/nips2012.html#ZhouPBM12.

[327] X. Wang et al., ‘Approximate Truth Discovery via Problem Scale Reduction’, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM ’15*, 2015, vol. 19-23-Oct-, pp. 503–512[Online]. Availablehttp://dx.doi.org/10.1145/2806416.2806444.

[328] B. Zhao et al., ‘A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration’, in *Proceedings of the VLDB Endowment (PVLDB)*, 2012, vol. 5, no. 6, pp. 550–561 [Online]. Available: 10.14778/2168651.2168656.

[329] B. Zhao and J. Han, ‘A Probabilistic Model for Estimating Real-valued Truth from Conflicting Sources’, in *Proceedings of the 10th International Workshop on Quality in Databases in conjunction with VLDB (QDB’12)*, 2012.

[330] L. Li et al., ‘Truth Discovery with Memory Network’, *Tsinghua Sci. Technol.*, vol. 22, no. 6, pp. 609–618, 2017 [Online]. Available: 10.23919/TST.2017.8195344.

[331] K. Broelemann et al., ‘Restricted Boltzmann Machines for Robust and Fast Latent Truth Discovery’, *CoRR*, vol. abs/1801.0, 2018[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1801.html#abs-1801-00283.

[332] N. Choudhary et al., ‘Neural Network Architecture for Credibility’, in *In the proceedings of CICLING 2018*, 2018, pp. 1–13[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1803.html#abs-1803-10547.

[333] Q. Li et al., *User Behaviors in Newsworthy Rumors : A Case Study of Twitter*, no. ICWSM. Association for the Advancement of Artificial Intelligence (www.aaai.org), 2016, pp. 627–630[Online]. Availablehttp://dblp.uni-trier.de/db/conf/icwsm/icwsm2016.html#LiLFNS16.

[334] C. Stab and I. Gurevych, ‘Identifying Argumentative Discourse Structures in Persuasive Essays’, in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, no. October, pp. 46–56 [Online]. Available: 10.3115/v1/d14-1006.

[335] M. Alshomary et al., ‘Target Inference in Argument Conclusion Generation’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, no. 1, pp. 4334–4345 [Online]. Available: 10.18653/v1/2020.acl-main.399.

[336] Y. Bilu and N. Slonim, ‘Claim synthesis via predicate recycling’, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, 2016, pp. 525–530 [Online]. Available: 10.18653/v1/p16-2085.

[337] L. Wang and W. Ling, ‘Neural network-based abstract generation for opinions and arguments’, in *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016, pp. 47–57 [Online]. Available: 10.18653/v1/n16-1007.

[338] C. Egan et al., ‘Summarising the Points Made in Online Political Debates’, in *Proceedings of the 3rd Workshop on Argument Mining (ArgMining2016)*, 2016, pp.

134–143 [Online]. Available: 10.18653/v1/w16-2816.

[339] C. Park et al., ‘Generating Sentential Arguments from Diverse Perspectives on Controversial Topic’, in *Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019, vol. November 4, pp. 56–65 [Online]. Available: 10.18653/v1/d19-5007.

[340] X. Hua and L. Wang, ‘Neural argument generation augmented with externally retrieved evidence’, in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, vol. 1, pp. 219–230 [Online]. Available: 10.18653/v1/p18-1021.

[341] X. Hua et al., ‘Argument generation with retrieval, planning, and realization’, in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2019, pp. 2661–2672 [Online]. Available: 10.18653/v1/p19-1255.

[342] H. Wachsmuth et al., ‘Argumentation Synthesis following Rhetorical Strategies’, in *Proceedings of the 27th International Conference on Computational Linguistics, Coling*, 2018, vol. August, pp. 3753–3765[Online]. Availablehttp://argumentation.bplaced.net/arguana-publications/papers/wachsmuth18b-coling.pdf.

[343] C. Hidey and K. McKeown, ‘Fixed that for you: Generating contrastive claims with semantic edits’, *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 1756–1767, 2019 [Online]. Available: 10.18653/v1/n19-1174.

[344] S. E. Toulmin, *The Uses of Argument*. Cambridge, UK: Cambridge University Press, 1958.

[345] I. Habernalt and I. Gurevych, ‘Which Argument is more Convincing? Analyzing and Predicting Convincingness of Web Arguments Using Bidirectional LSTM’, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2016, vol. 1, pp. 1589–1599[Online]. Availablehttps://www.aclweb.org/anthology/P16-1150.

[346] T. Berariu, ‘An Argumentation Framework for BDI Agents’, *Intell. Distrib. Comput. VII*, vol. 511, pp. 343–354, 2014[Online]. Availablehttps://doi.org/10.1007/978-3-319-01571-2_40.

[347] A. R. Panisson et al., ‘An Approach for Argumentation-based Reasoning Using Defeasible Logic in Multi-Agent Programming Languages’, *11th Int. Work. Argumentation Multiagent Syst.*, no. May, pp. 1–15, 2014[Online]. Availablehttp://www.mit.edu/~irahwan/argmas/argmas14/w12-06.pdf.

[348] A. R. Panisson and R. H. Bordini, ‘Knowledge Representation for Argumentation in Agent-Oriented Programming Languages’, *Proc. - 2016 5th Brazilian Conf. Intell. Syst. BRACIS 2016*, pp. 13–18, 2017 [Online]. Available: 10.1109/BRACIS.2016.014.

[349] P. E. Velmovitsky et al., ‘Practical reasoning in an argumentation-based decision BDI agent: A case study for participatory management of protected areas’, *Proc. Int. Conf. Softw. Eng. Knowl. Eng. SEKE*, pp. 527–530, 2017 [Online]. Available: 10.18293/SEKE2017-153.

[350] R. El Baff et al., ‘Computational Argumentation Synthesis as a Language Modeling Task’, in *INLG 2019 - 12th International Conference on Natural Language Generation, Proceedings of the Conference*, 2019, pp. 54–64 [Online]. Available: 10.18653/v1/w19-8607.

[351] S. E. Toulmin et al., *An Introduction to Reasoning*1984 ..

[352] V. Simaki et al., ‘A Two-step Procedure to Identify Lexical Elements of Stance Constructions in Discourse from Political Blogs’, *Corpora*, vol. 14, no. 3, pp. 379–405, 2019 [Online]. Available: 10.3366/cor.2019.0179.

[353] I. Habernal and I. Gurevych, ‘Exploiting Debate Portals for Semi-Supervised Argumentation Mining in User-Generated Web Discourse’, in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, no. September, pp. 2127–2137 [Online]. Available: 10.18653/v1/d15-1255.

[354] F. Boltuzic and J. Šnajder, ‘Fill the Gap! Analyzing Implicit Premises between Claims from Online Debates’, in *Proceedings of the 3rd Workshop on Argument Mining*, 2016, no. August, pp. 124–133 [Online]. Available: 10.18653/v1/w16-2815.

[355] K. Singh et al., ‘Ranking Warrants with Pairwise Preference Learning’, in *Proceedings of the 26th Annual Meeting of the Natural Language Processing Society (March 2020)*, 2020, no. C, pp. 776–779[Online]. Availablehttps://www.anlp.jp/proceedings/annual_meeting/2020/pdf_dir/P3-34.pdf.

[356] S. Chen et al., ‘PERSPECTROSCOPE: A Window to the World of Diverse Perspectives’, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2019, pp. 129–134[Online]. Availablehttps://www.aclweb.org/anthology/P19-3022.

[357] M. Risdal, (2016,), *Getting Real about Fake News*, kaggle. [Online]. Available: https://www.kaggle.com/mrisdal/fake-news. [Accessed: 08 Dec. 2020].

[358] R. Baly et al., ‘Predicting Factuality of Reporting and Bias of News Media Sources’, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020, pp. 3528–3539[Online]. Availablehttps://www.aclweb.org/anthology/D18-1389.

[359] B. Popken, (2018,), *Twitter deleted 200,000 Russian troll tweets. Read them here*, NBC News. [Online]. Available: https://www.nbcnews.com/tech/social-media/now-available-more-200-000-deleted-russian-troll-tweets-n844731. [Accessed: 08 Dec. 2020].

[360] C. Boididou et al., ‘Challenges of Computational Verification in Social Multimedia’, *WWW 2014 Companion - Proc. 23rd Int. Conf. World Wide Web*, pp. 743–748, 2014 [Online]. Available: 10.1145/2567948.2579323.

[361] A. Zubiaga et al., ‘Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media’, *arXiv Prepr. arXiv1610.07363*, pp. 1–20, 2016.

[362] Z. Jin et al., ‘Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs’, in *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, 2017, no. Fast Forward 3, pp. 795–816 [Online]. Available: 10.1145/3123266.3123454.

[363] K. Popat et al., ‘DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning’, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020, pp. 22–32 [Online]. Available: 10.18653/v1/d18-1003.

[364] K. Shu et al., ‘FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media’, *Big Data*, vol. 8, no. 3, pp. 171–188, 2020 [Online]. Available: 10.1089/big.2020.0062.

[365] I. Habernal et al., ‘The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants’, in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018, vol. 1, pp. 1930–1940[Online]. Availablehttps://arxiv.org/abs/1708.01425.

[366] I. Habernal et al., ‘SemEval-2018 Task 12: The Argument Reasoning Comprehension Task’, in *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, 2018, vol. June, pp. 763–772 [Online]. Available: 10.18653/v1/s18-1121.

[367] C. JAIN, ‘Detecting Fake News and Fake Reviews through Linguistic Styles’, Delhi Technological University, 2019.

[368] J. P. J. Kincaid et al., ‘Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel’, *Nav. Tech. Train. Command Millingt. TN Res. Branch*, pp. 8–75, 1975[Online]. Availablehttps://stars.library.ucf.edu/istlibrary/56%0A%0A.

[369] R. Gunning, *Technique of Clear Writing*. Toronto : McGraw-Hill, 1952.

[370] J. Graham et al., ‘Liberals and Conservatives Rely on Different Sets of Moral Foundations’, *J. Pers. Soc. Psychol.*, vol. 96, no. 5, pp. 1029–1046, 2009 [Online]. Available: 10.1037/a0015141.

[371] O. Levy and Y. Goldberg, ‘Dependency-Based Word Embeddings’, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2014, vol. 2, pp. 302–308[Online]. Availablehttps://www.aclweb.org/anthology/P14-2050.

[372] B. Ghanem et al., ‘An Emotional Analysis of False Information in Social Media and News Articles’, *ACM Trans. Internet Technol.*, vol. 20, no. 2, p. Article No.19, 2020[Online]. Availablehttps://doi.org/10.1145/3381750.

[373] C. Hashimoto et al., ‘Excitatory or Inhibitory: A New Semantic Orientation Extracts Contradiction and Causality from the Web’, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 619–630[Online]. Availablehttps://www.aclweb.org/anthology/D12-1057.

[374] R. Rinott et al., ‘Show Me Your Evidence – an Automatic Method for Context Dependent Evidence Detection’, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, no. September, pp. 440–450 [Online]. Available: 10.18653/v1/d15-1050.

[375] J. Liu et al., ‘Attention-based BiGRU-CNN for Chinese question classification’, *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2019[Online]. Availablehttps://doi.org/10.1007/s12652-019-01344-9.

[376] P. Isola et al., ‘Image-to-Image Translation with Conditional Adversarial Networks’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134 [Online]. Available: 10.1007/978-3-030-11009-3_37.

[377] D. Engin et al., ‘Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018, pp. 938–946[Online]. Availablehttp://arxiv.org/abs/1805.05308.

[378] S. Zhang et al., ‘Enhanced Text Matching Based on Semantic Transformation’, *IEEE Access*, vol. 8, no. February, pp. 30897–30904, 2020 [Online]. Available: 10.1109/ACCESS.2020.2973206.

[379] M. Lippi and P. Torroni, ‘Context-Independent Claim Detection for Argument Mining’, *Proc. Twenty-Fourth Int. Jt. Conf. Artif. Intell. (IJCAI 2015),* vol. January, pp. 185–191, 2015.

[380] J. Mueller et al., ‘Sequence to Better Sequence: Continuous Revision of Combinatorial Structures’, in *Proceedings of the 34th International Conference on Machine Learning - ICML,* 2017, vol. 5, no. 1, pp. 3900–3916.

[381] B. S. Neysiani and S. Morteza Babamir, ‘New Methodology for Contextual Features Usage in Duplicate Bug Reports Detection: Dimension Expansion based on Manhattan Distance Similarity of Topics’, *5th Int. Conf. Web Res. ICWR 2019,* no. 4, pp. 178–183, 2019 [Online]. Available: 10.1109/ICWR.2019.8765296.

[382] K. Guu et al., ‘Traversing Knowledge Graphs in Vector Space’, in *Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing - EMNLP,* 2015, pp. 318–327 [Online]. Available: 10.18653/v1/d15-1038.

[383] F. T. Al-Khawaldeh et al., ‘A Novel Model for Enhancing Fact-Checking’, in *Proceedings of the 2021 Computing Conference,* 2021, vol. 284, pp. 661–677[Online]. Availablehttps://doi.org/10.1007/978-3-030-80126-7_47.

[384] M. Heilman and K. Sagae, ‘Fast Rhetorical Structure Theory Discourse Parsing’, *CoRR, abs/1505.02425,* pp. 1–6, 2015[Online]. Availablehttp://arxiv.org/abs/1505.02425.

[385] J.-H. Oh et al., ‘Multi-Column Convolutional Neural Networks with Causality-Attention for Why-Question Answering’, in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 415–424[Online]. Availablehttps://doi.org/10.1145/3018661.3018737.

[386] A. Imani et al., ‘Deep Neural Networks for Query Expansion Using Word Embeddings’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* ECIR 2019., vol. 11438 LNCS, H. D. Azzopardi L., Stein B., Fuhr N., Mayr P., Hauff C., Ed. Springer, Cham, 2019, pp. 203–210[Online]. Availablehttps://doi.org/10.1007/978-3-030-15719-7_26.

[387] M. Esposito et al., ‘Hybrid Query ExpansionUsing Lexical Resources and Word Embeddings for Sentence Retrieval in Question Answering’, *Inf. Sci. (Ny).,* vol. 514, pp. 88–105, 2020[Online]. Availablehttps://doi.org/10.1016/j.ins.2019.12.002.

[388] N. Yusuf et al., ‘Enhancing Query Expansion Method Using Word Embedding’, in *Proceeding of the 9th IEEE International Conference on System Engineering and Technology*, 2019, vol. 6, pp. 21–24 [Online]. Available: 10.1109/ICSEngT.2019.8906317.

[389] H. K. Azad and A. Deepak, ‘A New Approach for Query Expansion Using Wikipedia and WordNet’, *Inf. Sci. (Ny).,* vol. 492, pp. 147–163, 2019[Online]. Availablehttps://doi.org/10.1016/j.ins.2019.04.019.

[390] C. dos Santos et al., ‘Attentive Pooling Networks’, *arXiv Prepr. arXiv1602.03609,* pp. 1–10, 2016[Online]. Availablehttp://arxiv.org/abs/1602.03609.

[391] K. Rao et al., ‘RL-CycleGan: Reinforcement Learning Aware simulation-to-real’, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2020, pp. 11154–11163 [Online]. Available: 10.1109/CVPR42600.2020.01117.

[392] B. Zhao et al., ‘Reconstructive Sequence-Graph Network for Video Summarization’, in *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2021, vol. 8828, no. c, pp. 1–10 [Online]. Available: 10.1109/TPAMI.2021.3072117.

[393] A. Ayoub et al., ‘Model-Based Reinforcement Learning with Value-Targeted Regression’, in *Proceedings of the 37th International Conference on Machine Learning,* 2020, vol. PMLR 119, pp. 463–474[Online]. Availablehttp://proceedings.mlr.press/v119/ayoub20a.html.

[394] P. Bachman and D. Precup, ‘Data Generation as Sequential Decision Making’, *Adv. Neural Inf. Process. Syst. 28*, vol. 2015-Janua, pp. 3249–3257, 2015.

[395] K. Cho et al., ‘Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation’, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP*, 2014, pp. 1724–1734 [Online]. Available: 10.3115/v1/D14-1179.

[396] F. T. Al-Khawaldeh et al., ‘RL-GAN Based Toulmin Argument’, *J. Appl. Sci. Comput. JASC*, vol. VII, no. III, pp. 106–120, 2020.

[397] J. Alammar, (2018,), *The Illustrated Transformer*. [Online]. Available: http://jalammar.github.io/illustrated-transformer/. [Accessed: 16 Nov. 2020].

[398] M. Lewis et al., ‘BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880 [Online]. Available: 10.18653/v1/2020.acl-main.703.

[399] H. Li et al., ‘Keywords-Guided Abstractive Sentence Summarization’, *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, vol. 34, no. AAAI-20 Technical Tracks 5, pp. 8196–8203, 2020 [Online]. Available: 10.1609/aaai.v34i05.6333.

[400] H. Van Hasselt, ‘Double Q-learning’, *Adv. Neural Inf. Process. Syst. 23 24th Annu. Conf. Neural Inf. Process. Syst. 2010, NIPS 2010*, pp. 2613–2621, 2010.

[401] Y. Keneshloo et al., ‘Deep Reinforcement Learning for Sequence-to-Sequence Models’, *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2469–2489, 2020 [Online]. Available: 10.1109/TNNLS.2019.2929141.

[402] A. Pourchot and O. Sigaud, ‘CEM-RL: Combining evolutionary and gradient-based methods for policy search’, *arXiv Prepr.*, vol. 2, no. 1, pp. 1–19, 2018.

[403] N. Hansen, ‘Lec2. Material--The CMA Evolution Strategy : A Tutorial’, *Arxiv e-prints,* vol. Aprile. p. 11, 2016[Online]. Availablehttps://arxiv.org/pdf/1604.00772.pdf.

[404] B. Keller et al., ‘Incorporating Copying Mechanism in Sequence-to-Sequence Learning’, *J. Med. Internet Res.*, vol. 16, no. 1, p. e8, 2014[Online]. Availablehttps://www.jmir.org/2014/1/e8/.

[405] R. Nallapati et al., ‘Abstractive text summarization using sequence-to-sequence RNNs and beyond’, in *The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL),* 2016, pp. 280–290 [Online]. Available: 10.18653/v1/k16-1028.

[406] D. P. Kingma and J. L. Ba, ‘ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION’, in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–15.

[407] K. Papineni et al., ‘BLEU: a Method for Automatic Evaluation of Machine Translation’, in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL),* 2002, no. JULY, pp. 311–318 [Online]. Available: 10.1002/andp.19223712302.

[408] C. W. Liu et al., ‘How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation’, in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing,*

Proceedings, 2016, pp. 2122–2132 [Online]. Available: 10.18653/v1/d16-1230.

[409] F. T. Al-Khawaldeh et al., 'Warrant Generation Through Deep Learning', *Comput. Sci. Inf. Technol. (CS IT),* vol. 11, no. 20, pp. 53–75, Nov. 2021 [Online]. Available: 10.5121/csit.2021.112005.

[410] J. Mueller and A. Thyagarajan, 'Siamese Recurrent Architectures for Learning Sentence Similarity', in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16),* 2016, no. November, pp. 2786–2792[Online]. Availablehttp://dblp.uni-trier.de/db/conf/aaai/aaai2016.html#MuellerT16.

[411] A. Celikyilmaz et al., 'Deep Communicating Agents for Abstractive Summarization', in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers),* 2018, vol. 1, no. June, pp. 1662–1675[Online]. Availablehttps://www.aclweb.org/anthology/N18-1150.

[412] R. Pasunuru and M. Bansal, 'Reinforced Video Captioning with Entailment Rewards', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* 2017, no. September, pp. 979–985[Online]. Availablehttps://www.aclweb.org/anthology/D17-1103.

[413] D. Bahdanau et al., 'Neural Machine Translation by Jointly Learning to Align and Translate', in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–15.

[414] Y. Miao et al., 'Neural Variational Inference for Text Processing', in *33rd International Conference on Machine Learning, ICML in Proceedings of Machine Learning Research*, 2016, vol. 4, no. Mcmc, pp. 2589–2600[Online]. Availableproceedings.mlr.press/v48/miao16.html.

[415] C.-Y. Lin, 'ROUGE: A Package for Automatic Evaluation of Summaries', in *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out (WAS 2004),* 2004, no. July, pp. 74–81[Online]. Availablehttps://www.aclweb.org/anthology/W04-1013.

[416] M. Liu et al., 'A Learning-Exploring Method to Generate Diverse Paraphrases with Multi-Objective Deep Reinforcement Learning', in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2310–2321[Online]. Availablehttps://aclanthology.org/2020.coling-main.209.

[417] C. Tao et al., 'Get The Point of My Utterance! Learning Towards Effective Responses with Multi-Head Attention Mechanism', in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18),* 2018, vol. 2018-July, no. July, pp. 4418–4424 [Online]. Available: 10.24963/ijcai.2018/614.

[418] V. Mnih et al., 'Human-level control through deep reinforcement learning', *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015 [Online]. Available: 10.1038/nature14236.

[419] P. Qin et al., 'Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 2018, pp. 2137–2147 [Online]. Available: 10.18653/v1/P18-1199.

[420] W. Xiong et al., 'DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* 2017, pp. 564–573 [Online]. Available: 10.18653/v1/D17-1060.

[421] Z. Li et al., 'Paraphrase Generation with Deep Reinforcement Learning', in

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3865–3878 [Online]. Available: 10.18653/v1/D18-1421.

[422] Z. Lin et al., 'A Structured Self-attentive Sentence Embedding', in *Proceedings of the ICLR Conference,* 2017[Online]. Availablehttp://arxiv.org/abs/1703.03130.

[423] J. Hasselqvist et al., 'Query-Based Abstractive Summarization Using Neural Networks', *CoRR, abs/1712.06100*, 2017[Online]. Availablehttp://dblp.uni-trier.de/db/journals/corr/corr1712.html#abs-1712-06100.

[424] B. Wang et al., 'Conditional Generative Adversarial Networks for Commonsense Machine Comprehension', in *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17*, 2017, pp. 4123–4129[Online]. Availablehttps://code.google.com/archive/p/word2vec/.

[425] H. Zhou et al., 'Commonsense Knowledge Aware Conversation Generation with Graph Attention', in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018, pp. 4623–4629[Online]. Availablehttps://doi.org/10.24963/ijcai.2018/643[Accessed: 21November2021].

[426] S. Gabriel et al., 'Paragraph-level Commonsense Transformers with Recurrent Memory', in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 14, pp. 12857–12865[Online]. Availablehttp://arxiv.org/abs/2010.01486[Accessed: 21November2021].

[427] G. Izacard and E. Grave, 'Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering', in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics,* 2021, vol. Main Volum, pp. 874–880 [Online]. Available: 10.18653/v1/2021.eacl-main.74.

[428] N. Reimers and I. Gurevych, 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processingand the 9th International Joint Conference on Natural Language Processing,* 2019, pp. 3982–3992[Online]. Availablehttps://github.com/UKPLab/.

[429] A. Radford et al., 'Language Models are Unsupervised Multitask Learners', *OpenAI blog,* vol. 1, no. 8, p. 9, Feb. 2019[Online]. Availablehttps://github.com/codelucas/newspaper.

[430] Y. Zhu et al., 'Texygen: A benchmarking platform for text generation models', in *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, 2018, pp. 1097–1100 [Online]. Available: 10.1145/3209978.3210080.

[431] L. Xiao et al., 'Gated Multi-Task Network for Text Classification', in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers),* 2018, vol. 2, pp. 726–731 [Online]. Available: 10.18653/v1/n18-2114.

[432] A. See et al., 'Get To The Point: Summarization with Pointer-Generator Networks', in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers),* 2017, vol. 1, pp. 1073–1083 [Online]. Available: 10.18653/v1/P17-1099.

[433] W. T. Hsu et al., 'A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss', in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, vol. 1, pp. 132–141 [Online]. Available: 10.18653/v1/p18-1013.

[434] Q. Zhou et al., 'Selective Encoding for Abstractive Sentence Summarization', in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics,*

Proceedings of the Conference (Long Papers), 2017, vol. 1, pp. 1095–1104 [Online]. Available: 10.18653/v1/P17-1101.

[435] Y. C. Chen and M. Bansal, ‘Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting’, in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, vol. 1, no. 2017, pp. 675–686 [Online]. Available: 10.18653/v1/p18-1063.

[436] X. Chen et al., ‘Iterative Document Representation Learning Towards Summarization with Polishing’, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020, pp. 4088–4097[Online]. Availablehttp://dblp.uni-trier.de/db/conf/emnlp/emnlp2018.html#ChenGTSZY18.

[437] K. M. Hermann et al., ‘Teaching Machines to Read and Comprehend’, in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS’15)*, 2015, pp. 1693–1701[Online]. Availablehttp://dblp.uni-trier.de/db/conf/nips/nips2015.html#HermannKGEKSB15.

[438] G. Heinrich, ‘Parameter Estimation for Text Analysis’, 2009[Online]. Availablehttp://www.arbylon.net/publications/text-est.pdf.

[439] T. M. COVER, *Elements of Information Theory*. John Wiley & Sons, 1999.

[440] A. Mehri et al., ‘Word Ranking in a Single Document by Jensen-Shannon Divergence’, *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 379, no. 28–29, pp. 1627–1632, 2015[Online]. Availablehttp://dx.doi.org/10.1016/j.physleta.2015.04.030.

[441] R. J. Gallagher et al., ‘Divergent discourse between protests and counter-protests: #BlackLivesMatter and #AllLivesMatter’, *PLoS One*, vol. 13, no. 4, pp. 1–23, 2018 [Online]. Available: 10.1371/journal.pone.0195644.

[442] S. Gao et al., ‘Abstractive Text Summarization by Incorporating Reader Comments’, *33rd AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 6399–6406, 2019[Online]. Availablehttps://doi.org/10.1609/aaai.v33i01.33016399.

[443] P. Nema et al., ‘Diversity driven Attention Model for Query-based Abstractive Summarization’, in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, vol. 1, pp. 1063–1072 [Online]. Available: 10.18653/v1/P17-1098.

[444] S. M. Mohammad et al., ‘SemEval 2016 Task 6 Detecting Stance in Tweets’, in *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval 2016*, 2016, pp. 31–41 [Online]. Available: 10.18653/v1/s16-1003.

[445] K. Popat et al., ‘STANCY: Stance Classification Based on Consistency Cues’, 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)*, 2019, pp. 6413–6418[Online]. Availablehttps://www.aclweb.org/anthology/D19-1675.