Øystein Løndal Nilsen

From Uni-Modal to Multi-Modal Fake News Detection

The Impact of Visual Cues on Detection Performance

Master's thesis in Computer Science
Supervisor: Özlem Özgöbek
June 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Øystein Løndal Nilsen

From Uni-Modal to Multi-Modal Fake News Detection

The Impact of Visual Cues on Detection Performance

Master's thesis in Computer Science
Supervisor: Özlem Özgöbek
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

ABSTRACT

The proliferation of misinformation, popularly known as fake news, on social media is a pressing concern due to its potential impact on crucial events like elections and global emergencies. Existing detection methods primarily focus on text-based news, often neglecting the value of integrating multiple modes of information, particularly visual cues, which previous studies have found to be highly effective. This thesis delves into this relatively unexplored domain by introducing a sophisticated multi-modal framework called the Image-enhanced Knowledge-Aware Hierarchical Attention Network (I-KAHAN), which builds upon a state-of-the-art uni-modal fake news detection system. This framework effectively combines textual and visual attributes to enhance the detection of fake news.

To effectively incorporate visual elements, various techniques were experimented with to determine the optimal combination of image embedding, dimensionality reduction, and feature combination techniques. The most promising methods, determined through experimentation, include the use of CLIP for image embedding and a novel dimensionality reduction method called IHAN. The experiments revealed that CLIP-based image embeddings, pooling-based dimensionality reduction, and concatenation-based feature fusion yielded the best performance. Additionally, the novel dimensionality reduction method IHAN showed excellent performance, indicating its significant potential. Furthermore, the baseline neural network classifier was compared to a version with an additional hidden layer, aiming to enhance representational power to accommodate the complexity introduced by adding the visual feature. Surprisingly, the shallow classifier outperformed its more complex counterpart in almost all the cases, providing unexpected insights.

To address concerns regarding data quality, enhancements were implemented in the FakeNewsNet dataset collection process, leading to noticeable improvements. These enhancements, collectively known as FakeNewsNet+, significantly boosted the performance, with as much as 10% in some circumstances.

I-KAHAN outperformed the baseline uni-modal model across all metrics, demonstrating an improvement of approximately 1% and 3% on the GossipCop and PolitiFact datasets, respectively. These results reinforce the findings of previous research, which emphasize the significance of visual attributes as crucial cues for distinguishing between real and fake news. While this study significantly advances the field of fake news detection by introducing an innovative model and uncovering valuable insights, it acknowledges certain limitations. Concerns regarding the model's generalizability and ethical implications, such as potential biases and misuse, emphasize the need for careful application and ongoing refinements. Despite these challenges, this study sheds light on the promising future of multi-modal fake news detection and underscores the necessity for continued research in the ongoing battle against misinformation.

SAMMENDRAG

Spredningen av misinformasjon, også kjent som falske nyheter, på sosiale medier utgjør en alvorlig trussel. Det kan påvirke avgjørende hendelser som valg og globale kriser. Eksisterende metoder for deteksjon av falske nyheter fokuserer hovedsakelig på individuelle nyhetsegenskaper, som for eksempel tekstinnhold, og overser i mange tilfeller betydningen av å integrere flere informasjonsformer. Det er spesielt verdt å merke seg at visuelle elementer har vist seg å være effektive for å skille mellom ekte og falske nyheter gjennom tidligere forsking. Denne masteroppgaven utforsker dette relativt uutforskede området ved å introdusere et omfattende flermodalt rammeverk kalt Image-enhanced Knowledge-Aware Hierarchical Attention Network (I-KAHAN), som bygger videre på et sofistikert unimodalt system for å oppdage falske nyheter. Dette rammeverket tar i bruk både tekstlige og visuelle elementer fra nyheter, med mål om å overgå det unimodale systemets klassifiseringsytelse.

For en optimal integrering av visuelle elementer, ble det utført omfattende eksperimentering med ulike teknikker for numerisk bilderepresentasjon, dimensjonalitetsreduksjon, og aggregering av elementer. Gjennom eksperimenteringen identifiserte vi en rekke lovende metoder, inkludert CLIP for bildeinnkapsling og en egenutviklet dimensjonalitetsreduksjonsmetode kalt IHAN. Eksperimentene viste at bildeinnkapsling basert på CLIP, dimensjonalitetsreduksjon via pooling, og aggregering via konkatinering resulterte i den best ytelsen. IHAN demonstrerte dessuten utmerket ytelse, noe som antyder dens store potensial. Videre sammenlignet vi den originale nevrale nettverksbaserte klassifikatoren med en modifisert versjon med et ekstra skjult lag. Målet med denne endringen var å forbedre representasjonsevnen og håndtere den økte kompleksiteten fra introduksjonen av et ekstra nyhetsattributt. Imidlertid overgikk den grunnleggende klassifikatoren den mer komplekse varianten i de fleste tilfellene.

Datakvalitet var en stor bekymring, så det ble i tillegg implementert forbedringer i datainnsamlingsprosessen for datasettene. Dette utgjorde merkbare forbedringer, hvor den forbedrede prosessen, kalt for FakeNewsNet+, førte til en betydelig ytelsesøkning med opptil 10% i enkelte tilfeller.

I-KAHAN overgår det grunnleggende unimodale systemet på alle metrikker og viser en forbedring på omtrent 1% og 3% for henholdsvis GossipCop og PolitiFact datasettene. Disse resultatene støtter tidligere forskningsfunn som understreker viktigheten av visuelle attributter. Selv om denne studien bidrar betydelig til feltet for deteksjon av falske nyheter gjennom en innovativ modell og omfattende eksperimentering, fins det viktige begrensninger å anerkjenne. Bekymringer rundt modellens generaliserbarhet og etiske implikasjoner, som potensielle skjevheter og misbruk, understreker behovet for forsiktig bruk og kontinuerlige forbedringer. Til tross for disse utfordringene, fremhever denne studien den lovende fremtiden for flermodal nyhetsklassifisering, og poengterer samtidig behovet for vedvarende forskning i kampen mot misinformasjon.

PREFACE

This thesis is being submitted to the Norwegian University of Science and Technology (NTNU) as part of the TDT4900 - Master Thesis course, building upon my prior work in the TDT4501 - Specialization Project course [1]. It has been supervised by Associate Professor Özlem Özgöbek from NTNU's Department of Computer and Information Science (IDI). I am immensely grateful to Özlem Özgöbek for her guidance in the research process and valuable feedback. I would also like to express my heartfelt appreciation to Postdoctoral Researcher Eniafe F. Ayetiran, also from IDI, for the valuable discussions and feedback provided throughout the thesis work.

Table of Contents

List of Figures

List of Tables

CHAPTER
ONE

INTRODUCTION

In an age where information is disseminated at an unprecedented pace, the task of identifying truth from falsehood has become both critical and challenging. Social media platforms, while reshaping the landscape of news consumption, have concurrently become breeding grounds for misinformation, raising vital questions about the integrity of the information ecosystem.

This introductory chapter sets the stage for an in-depth exploration of the complex and increasingly important field of fake news detection. It outlines the contemporary relevance of the issue and the need for effective countermeasures. To address this pressing problem, this study focuses on augmenting existing detection systems by integrating visual elements from news content, a promising but somewhat unexplored area in fake news detection research.

The chapter begins by discussing the context and motivation, followed by an outline of the problems faced in the field of fake news detection. It then frames the goal of the thesis and the research questions that guide this study. A brief overview of the research methods and the contributions of this thesis is also provided. The chapter concludes by presenting an outline of the subsequent chapters in the thesis, guiding readers through the subsequent exploration of this important subject.

1.1 Background and Motivation

Over recent years, the Internet has drastically evolved and become a cornerstone in our daily lives, predominantly in how we consume news. This transformation is attributed to the rise of social media platforms that enable prompt information dissemination among a vast user base. Twitter, a key participant in the realm of social media, reports an impressive count of 237.8 million active users, with nearly 500 million tweets being generated every day as of March 2023[1]. However, these platforms also exhibit significant challenges. For instance, the rapid spread of false information poses a pressing concern,

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

made more complex by the difficulties surrounding the verification of content [2].

A series of momentous events, including the 2016 US presidential election, have amplified the severity of the misinformation problem. In particular, the 2016 election saw the widespread sharing and belief in fake news, with pro-Trump fake stories shared on Facebook 30 million times, significantly more than pro-Clinton fake stories [3]. This phenomenon underscores the persuasive power of misinformation in shaping public opinion and influencing significant events. Following these, the COVID-19 pandemic, and the Russian invasion of Ukraine, further exacerbated the issue. The World Health Organization (WHO) Director-General characterized the vast spread of fake news during the pandemic as not only battling a disease but also an *infodemic*[2]. To illustrate, approximately half of the participants in a 2020 study on the misinformation about COVID-19 believed the pandemic was a worldwide conspiracy [4]. Moreover, misinformation, often termed as *fake news*, associated with the pandemic has been observed to amplify anxiety among individuals, subsequently impacting their overall health and well-being [5]. The numerous instances of fabricated videos and images emerging from the Russia-Ukraine conflict exemplify the exploitation of misinformation as a strategic tool in warfare, aimed at controlling the narrative [6].

In response to this issue, social media platforms have adopted strategies such as appending warnings to content suspected of being false. While these warnings demonstrate potential in curbing the acceptance of false content, their effect remains limited. The challenge is further complicated by the rise of advanced artificial intelligence technologies, notably large language models (LLMs), which can convincingly generate human-like texts, and could be exploited to create deceptive content [7]. Current approaches encounter difficulties in consistently identifying and removing misinformation, primarily due to the dynamic nature of social media, the subtlety of false narratives, and the continuous adaptation of new strategies by the producers of misinformation [8]. Therefore, the development of more sophisticated tools for detecting and eradicating fake news is crucial.

1.2 Problem Outline

The increasingly complex landscape of misinformation, or what is defined as fake news in this thesis, warrants urgent and comprehensive research into effective detection strategies [3]. In this study, the narrow definition by Allcott and Gentzkow, characterizing fake news as *a news article that is intentionally and verifiably false* [3], is adopted. Essentially, fake news is not about inaccuracies from high-quality sources that are promptly corrected. Instead, it pertains to content intentionally designed to deceive and mislead readers. Throughout this thesis, the terms misinformation and fake news are used interchangeably, adhering to the above definition.

Existing methodologies for fake news detection vary, encompassing uni-modal systems like KAHAN [9], which depend on a single type of news attribute, and multi-modal

[2] `https://www.who.int/director-general/speeches/detail/munich-security-conference`

systems like FakeMine [10] and SAFE [11], which utilize multiple news attributes to detect fake news. Leveraging multiple modalities can offer a comprehensive view of the news content, enhancing the potential for accurate fake news detection.

News attributes like images, in particular, are an essential component of news content. Most news articles include images related to the topic, providing additional context. Interestingly, research suggests that there are notable differences between the images used in real and fake news. For instance, Cao et al. [12] found that images in fake news tend to be more visually captivating and provocative. Similar results have been found by other researchers like Zhou et al. [13] and Segura-Bedmar et al. [14], who showed that the use of visuals can positively affect news classification performance. This implies that images could be an important tool in detecting fake news, supplementing the information derived from other modalities such as text.

Despite the inclusion of images in past research on multi-modal fake news detection, the optimal methods for integrating visual information remain elusive. Various techniques have been deployed in this research domain, including deep learning methods. However, their effectiveness may vary depending on the specific context and characteristics of the fake news. This thesis aims to fill the research gap by integrating and comparing multiple competing techniques on diverse datasets within a novel fake news detection framework. This framework intends to enhance a state-of-the-art uni-modal detection system by integrating visual information, thereby transforming it into a multi-modal system. Through evaluating the impact of visual attributes on system performance and comparing it with state-of-the-art multi-modal detection systems, this research seeks to shed light on effective ways of incorporating visual information. These insights hold the potential to significantly advance the field of fake news detection.

1.3 Goal and Research Questions

In view of the challenges outlined above, the primary objective of this thesis is encapsulated in the below statement.

Goal *This thesis seeks to examine the effects of incorporating visual attributes of news into a fake news detection system on its classification performance. Furthermore, it aims to evaluate and compare different image integration techniques and assess the resulting system's performance against existing multi-modal fake news detection systems.*

To accomplish this goal, the research will address the following research questions: **RQ1**, **RQ2**, and **RQ3**.

RQ1 *What techniques are most effective for incorporating visual elements of news into a multi-modal fake news detection system, thereby improving its classification performance?*

RQ2 *How significantly does the integration of visual attributes into a fake news detection system influence its classification performance?*

RQ3 *How does the classification performance of the developed multi-modal fake news detection system, which includes visual elements, compare with existing state-of-the-art multi-modal systems?*

1.4 Research Method and Contributions

In this study, a quantitative, experimental approach was employed to enhance the capabilities of an existing fake news detection model. The research broadened the model's scope by incorporating an additional feature, namely the images associated with news items. This required the implementation of components capable of numerically representing these images and integrating them with the textual features. To maximize the efficacy of this integration, various image embedding techniques, dimensionality reduction methods, and feature fusion techniques were compared and evaluated.

The process and findings of this research constitute major contributions to the understanding and advancement of multi-modal fake news detection. These contributions are best encapsulated in three key areas, seen below.

Framework A novel framework for multi-modal fake news detection has been developed, integrating not only textual information but also images and external knowledge. This comprehensive approach provides a more nuanced and precise mechanism for news classification.

Techniques An extensive comparative analysis of various image embedding, dimensionality reduction, and feature fusion techniques has been conducted to optimize image integration into the detection system. This includes the proposal of a novel dimensionality reduction method, as well as an exploration of the potential of attention for improved image representations.

Data Collection A refined data collection process was implemented, resulting in improved data quality. A thorough analysis of the impact of this improved data on detection performance also forms a significant part of this research.

1.5 Thesis Outline

The thesis is structured into nine chapters, with the last eight summarized and presented below.

Chapter 2 offers a theoretical background on fake news, machine learning, deep learning, attention mechanisms, embeddings, knowledge extraction, and fake news detection.

Chapter 3 examines related work in the field of uni-modal and multi-modal fake news detection, emphasizing the model our research builds upon.

Chapter 4 discusses the datasets used, including details on data cleaning, preparation, presentation, and visualization.

Chapter 5 outlines the research method, detailing the architecture of our novel framework, with the various image embedding techniques, dimensionality reduction methods, and feature fusion techniques implemented.

Chapter 6 provides a detailed overview of the experimental design, including tools and technologies used, the experimental setup, and the execution of the experiments.

Chapter 7 presents the results, providing a comparative analysis of different techniques, the impact of improved data collection, and the performance of the extended model.

Chapter 8 evaluates the results, discussing the performance of the model, the impact of different techniques and improved data collection on performance. It also considers the strengths, limitations, and ethical implications of the proposed model.

Finally, **Chapter 9** draws conclusions on the research based on whether the research questions have been adequately met. It also suggests future work and research directions in the field of multi-modal fake news detection.

CHAPTER
TWO

THEORETICAL BACKGROUND

This chapter provides a comprehensive overview of the theoretical foundations essential for studying the detection of fake news. It aims to present the key concepts and theories related to these systems in a clear and organized manner.

To begin, the chapter defines and examines the concept of fake news, elucidating its characteristics and the motivations behind its creation.

The next section explores various aspects of machine learning, specifically focusing on supervised, unsupervised, and self-supervised learning. Additionally, it delves into the field of deep learning, discussing feed-forward, deep, and convolutional neural networks. Moreover, recurrent neural networks, including long short-term memory, gated recurrent units, and bi-directional recurrent neural networks, are covered in detail.

Particular emphasis is given to attention mechanisms, which play a vital role in allowing models to assign importance to different parts of the input sequence. This section highlights self-attention, multi-head attention, co-attention, and hierarchical attention networks.

Furthermore, the chapter explains the concept of embeddings and their role in representing data in a lower-dimensional space for computational purposes. It explores how embeddings are applied in the context of fake news detection. Additionally, the chapter delves into various approaches to extracting knowledge, such as entity extraction, linking, and claim identification.

2.1 Fake News

> *a news article that is intentionally and verifiably false*
>
> *Allcott and Gentzkow*

The realm of fake news is a complex landscape that requires careful exploration for a comprehensive understanding. Definitions of fake news vary, with Allcott and Gentzkow

[3] characterizing it as a news article purposefully fabricated to deceive, and whose falsehood can be verified. Essentially, fake news contains information that is both intentionally deceptive and verifiable as false [15]. However, broader interpretations may encompass all deceptive news, including fabrications, hoaxes, and satire. For clarity and precision in this discussion, we will adhere to the more narrow definition provided by Allcott and Gentzkow. By following this definition, fake news cannot be confused with articles of high journalistic quality from trusted sources that are quickly corrected if reports of its inaccuracy arise, but rather content intentionally designed to provoke and mislead [16].

2.1.1 Categorization and Relationships

The Internet swarms with misleading information of various kinds. Distinguishing between the different types, particularly misinformation and disinformation, is crucial to understanding this complex landscape [17]. Misinformation refers to potentially misleading information without any intent of deception, while disinformation involves the intentional propagation of inaccurate information. Fake news, according to our adopted definition, forms a subset of disinformation. Several subcategories of misinformation exist, such as rumors and other forms of misleading content [2]. Rumors are unverified pieces of information shared online and can be divided into long-standing rumors and breaking news rumors. Clickbait and social spam represent other forms of misinformation. Further, fake news itself encompasses serious fabrications, large-scale hoaxes, and humorous fakes. Serious fabrications pertain to the aggressive dissemination of false information, large-scale hoaxes present fabricated stories as authentic news, and humorous fakes contain satirical content masquerading as news. The relationships among these categories are depicted in Figure 2.1.

2.1.2 Distinguishing Factors of Fake News

To differentiate between fake news and real news, we can consider several key characteristics, particularly in terms of the textual and visual content.

Textual Characteristics Real news articles are typically grounded in facts and objectivity. They present information from multiple perspectives, offering a balanced view. On the other hand, fake news articles tend to be subjective and biased, often representing a single viewpoint [18]. In terms of style, fake news articles often prioritize sensationalism and evoke emotional responses.

Visual Characteristics Fake news frequently utilizes manipulated or entirely fabricated images, as highlighted by [12]. These images may involve elements that have been photoshopped, images taken out of context, or completely computer-generated visuals also known as deep fakes[1]. Another clue lies in the visual style of an image, including aspects like color distribution, texture, and shape. Fake news images may

[1] `https://www.merriam-webster.com/dictionary/deepfake`

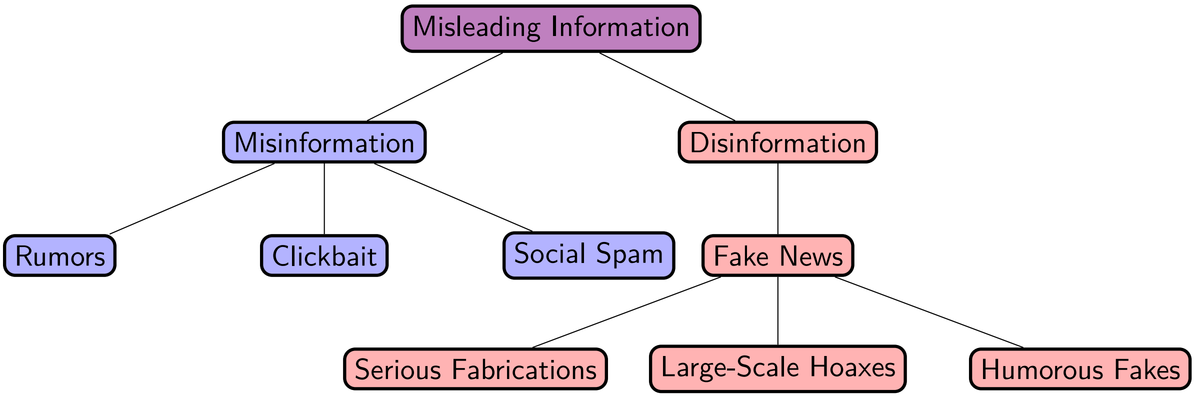


Figure 2.1: A visually differentiated tree diagram elucidating the various categories and subcategories of misleading information. Misinformation and disinformation are two main branches, with the latter encompassing fake news. Furthermore, fake news is segmented into serious fabrications, large-scale hoaxes, and humorous fakes. The diagram showcases the complex structure of false information distribution online, helping to understand their interrelations and individual characteristics. The color coding is utilized for clear distinction: the root node is half blue and half red, representing the overarching concept of *Misleading Information*, blue for *Misinformation* that signifies incorrect or misleading information presented without malicious intent, and red for *Disinformation*, including *Fake News*, which involves intentional misinformation with a purpose to deceive.

exhibit a distinct visual style that differs from authentic news images. Furthermore, examining the metadata of an image, such as the camera model, date, and location, can also aid in detecting fake news. Fake news images often exhibit inconsistent or missing metadata. [12] further elaborates that there are also semantic differences between real and fake news images, stating that the latter images are more visually captivating and provocative.

2.1.3 Motives Behind Fake News

According to the research conducted by [3], the production and dissemination of fake news can be traced back to two primary motivations that drive its creation. The first motivation is centered around financial gains, as online platforms generate revenue through advertisements based on the number of clicks they receive. This profit-driven incentive has become increasingly prevalent, with the 2016 US election serving as a notable example. During that time, numerous fabricated stories were intentionally produced to favor different candidates, solely driven by the desire for monetary gain.

The second motivation behind the production of fake news is ideological in nature. In this context, false narratives are strategically crafted to serve a specific agenda, often aiming to advance the interests of a particular candidate or cause. These deliberate distortions of information are designed to manipulate public opinion, sway political discourse, and shape the narrative surrounding certain issues. By fabricating and disseminating false stories, purveyors of fake news seek to influence and mold public perception in alignment with their ideological goals.

2.2 Machine Learning

The contemporary technological landscape is witnessing an increasingly prevalent utilization of Machine Learning (ML), a branch of computer science that focuses on algorithms and methods to solve complex problems that are hard to handle with traditional programming [19]. The reach of ML is vast, impacting everything from the way we search the web to how our smartphones function [20]. The versatility of ML also means it is used in various tasks, like recognizing objects, transcribing speech to text, and creating personalized suggestions.

Supervised and unsupervised learning constitute the two main paradigms in machine learning. Supervised learning, as described by LeCun *et al.*, involves learning correlations between data features and labels using a labeled dataset, enabling label prediction for unseen data. Unsupervised learning, conversely, deals with finding structures and patterns within unlabeled data, commonly used for tasks like data clustering [21]. An additional learning paradigm, self-supervised learning, leverages the advantages of both. According to [22], this approach creates *labels* from the data itself for guiding the learning process. Although the learning is unsupervised, it uses these derived labels, somewhat resembling supervised learning.

2.2.1 Supervised Learning

The fundamental components of supervised ML are illustrated in Figure 2.2. The depicted model consists of two phases: learning and evaluation, followed by real-world predictions. In the learning phase, the ML algorithm is trained on a dataset that represents the target domain. Subsequently, the algorithm is evaluated on the same dataset to assess its quality and performance. If the evaluation reveals poor performance, adjustments can be made to the algorithm, dataset, or learning process. This evaluation phase provides insights into the model's expected performance when deployed and serves as the basis for iterative improvements. The dataset is typically split into two parts: a training set and a test set. The usual split ratio is 80/20, with the majority allocated to the training set. Approximately 4/5 of the dataset is used for training, while the remaining 1/5 is used to evaluate the model by comparing predicted labels with true labels. To enhance the robustness of evaluation, *k*-fold cross-validation can be employed. This technique involves training the model on *k-1* subsets and evaluating it on the remaining subset. The process is repeated *k* times, with each subset serving as the evaluation set once. The performance scores obtained in each iteration are averaged to estimate the model's overall performance. After the learning and evaluation phases, the learned model is utilized to make predictions on real-world data, allowing the system to provide high-quality predictions for unseen instances.

Some supervised learning methods include Decision Tree, Naïve Bayes, and Support Vector Machine (SVM), each with distinct characteristics and applications, optimizing classification and regression tasks [21].

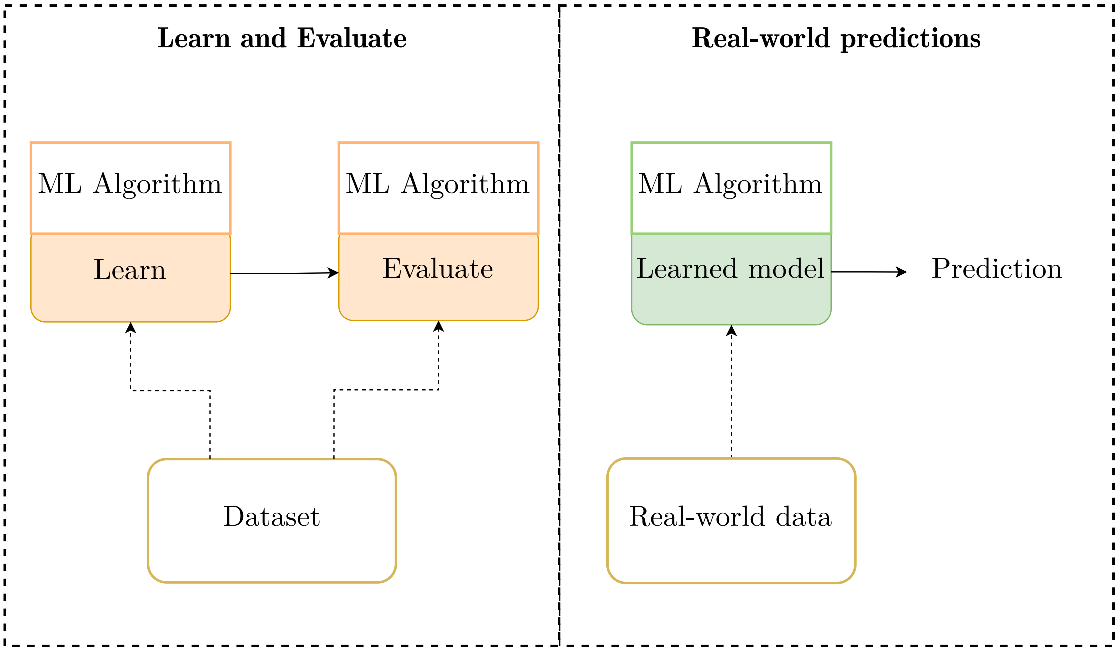


Figure 2.2: Illustration of the different phases in supervised machine learning. The figure is divided into two parts: the left side showcases the learning and evaluation process, while the right side demonstrates the application of the learned model to real-world predictions. Solid arrows represent the sequential order of events within each phase, while dashed arrows indicate the corresponding data flow. In the learning phase, the events *Learn* and *Evaluate* are highlighted in orange, indicating their association with the unfinished model, while the event on the right-hand side depicts the completed, learned model.

2.2.2 Unsupervised Learning

Unsupervised learning, another ML subset, involves pattern identification and structure discernment within unlabeled data. Unlike supervised learning, where a machine learns from labeled data, unsupervised learning focuses on comprehending the data's inherent structure. The model processes sequences of inputs and generates models that encapsulate the data's significant information.

Unsupervised learning's key strategies include clustering, grouping similar data points, and dimensionality reduction, extracting pertinent data features. Such methods facilitate pattern detection in data that transcend random noise, facilitating predictions, data representation, information dissemination, and decision making [23]. Algorithms such as k-means and k-nearest neighbors (kNN) are often employed for data clustering and unsupervised learning tasks [21, 24].

2.2.3 Self-supervised Learning

Self-supervised learning (SSL) is a machine learning method that harnesses large amounts of unlabeled data, standing as a notable alternative to traditional supervised and unsupervised learning [22]. SSL creates pretext tasks from the data, aiding models to learn informative representations beneficial across different domains, such as natural language processing and computer vision.

Pretext tasks in SSL might involve predicting the context of a masked word in a sentence or missing parts of an image. This form of learning helps models to grasp intrinsic relationships within data without explicit labels, making SSL models effective for various tasks like language translation or image generation.

SSL's main advantage is its ability to learn from extensive unlabeled data, yielding general and robust representations applicable across multiple tasks. It is especially useful in areas like healthcare, where labeled data is scarce, or the specific task is undefined [22].

SSL traces its roots back to early deep learning experiments, with techniques like Recurrent Neural Networks (RNN) and Transformers being prominent examples. These techniques are further discussed and elaborated later in this chapter. Modern SSL methods can be grouped into four families: Deep Metric Learning, Self-Distillation, Canonical Correlation Analysis, and Masked Image Modeling [22].

In essence, SSL provides a way to effectively utilize large volumes of unlabeled data, offering notable benefits in terms of generalization, robustness, and applicability to diverse tasks.

2.3 Deep Learning

Traditional machine learning approaches have inherent limitations in processing raw data, necessitating meticulous data engineering to render it suitable for these methods [20]. Deep learning overcomes this challenge by allowing deep techniques to autonomously learn representations of the raw data without human intervention. Most deep learning techniques are built upon the concept of the Perceptron, which originated from the 1958 paper by Frank Rosenblatt [25]. The Perceptron was designed as a hypothetical nervous system or machine to capture the intrinsic attributes of intelligent systems. At its core, the perceptron is a network of interconnected units that process stimuli and generate responses based on learned connections and activation thresholds. Minsky and Papert further developed this idea in 1969 by introducing the two-layered perceptron, capable of learning more complex representations than the original single-layered perceptron due to the increase in parameters [26]. However, the two-layered perceptron could only adequately approximate simple functions, later addressed by Hornik with the introduction of the multilayer perceptron in 1988, capable of representing any measurable function.

The perceptron and its descendants are categorized as feed-forward networks, as information flows from input through the activations of the layer(s) to the output. Another branch of neural networks, as explained in [27], comprises recurrent networks where feedback connections are created using loops.

2.3.1 Feed-forward Neural Networks

Gardner and Dorling provide an explanation of the structure of feed-forward neural networks [28]. These networks are composed of fully-connected neurons, also known as nodes, with weighted connections. A fully-connected architecture means that each neuron in one layer is connected to every neuron in the subsequent layer. The output signal of a neuron is obtained by applying a non-linear function to the sum of the weighted input signals. This process is repeated for each neuron in each subsequent layer, with the output signals of all neurons in the current layer serving as input to every neuron in the next layer. This iterative process continues until reaching the output layer, which is the final layer of the network.

The training of feed-forward neural networks follows a supervised approach, where the weights (parameters) are adjusted based on the error value between the predicted and true output values [20]. The adjustment of weights is performed using computed gradients by the learning algorithm. These gradients indicate the amount of increase or decrease in the error term if a specific weight is adjusted. The process of adjusting the weights based on the computed gradients is known as backpropagation, as the adjustments are propagated backward through the network.

Deep Neural Networks

Deep neural networks (DNNs) are an extension of feed-forward neural networks that incorporate multiple hidden layers between the input and output layers. Each hidden layer consists of a significant number of interconnected neurons, making DNNs capable of capturing complex patterns and representations. Figure 2.3 presents a visual representation of a generic DNN architecture, showcasing the input layer, output layer, and multiple hidden layers. The number of hidden layers and the number of neurons in each layer can vary depending on the complexity of the task and the available resources. DNNs with deeper architectures and larger numbers of neurons have shown improved performance

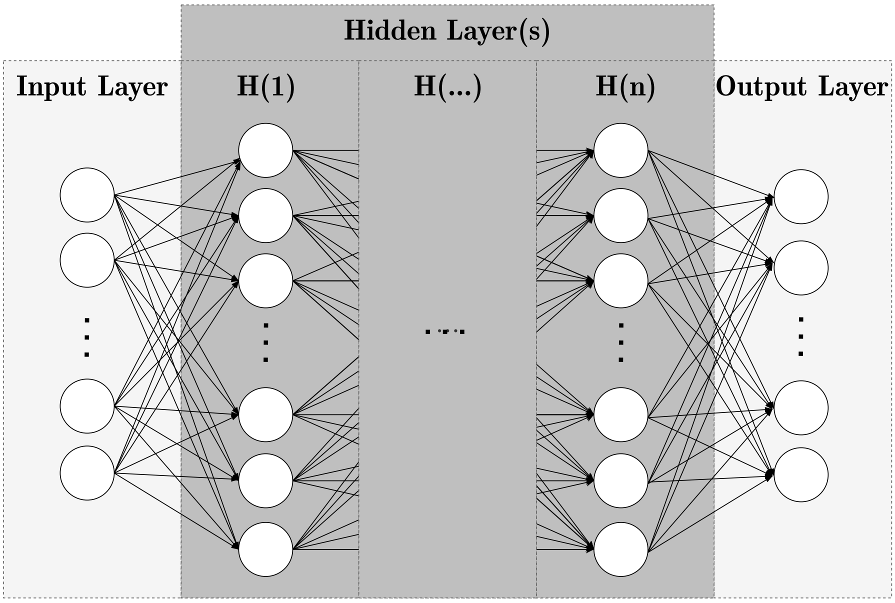


Figure 2.3: A visual representation of a generic deep neural network (DNN). The network consists of an input layer, an output layer, and multiple hidden layers. The presence of an empty middle hidden layer in the illustration highlights the possibility of incorporating a lot of hidden layers in DNNs. Similarly, the three dots between the top and bottom neurons of each layer indicate that a substantial number of neurons can be present within each layer. The arrows depict the flow of information from left to right, representing the feed-forward nature of the network

in various domains, such as image recognition, natural language processing, and speech recognition.

Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of deep neural network (DNN) designed specifically for processing structured grid data such as images and audio spectrograms, by considering the locality and stationarity of the input data [20]. While a DNN connects each neuron to all neurons in the next layer, a CNN only connects each neuron to a subset of neurons in the next layer, allowing it to capture local patterns.

Figure 2.4 depicts a model of a simple CNN. Here, an input image is sent to the network composed of multiple convolutional layers followed by pooling. The output is then flattened and passed to fully-connected layers, finally resulting in softmax probabilities. The darker color of output neurons signifies higher probabilities. Given the input image is of a dog, the darkest neuron corresponds to the class of a dog.

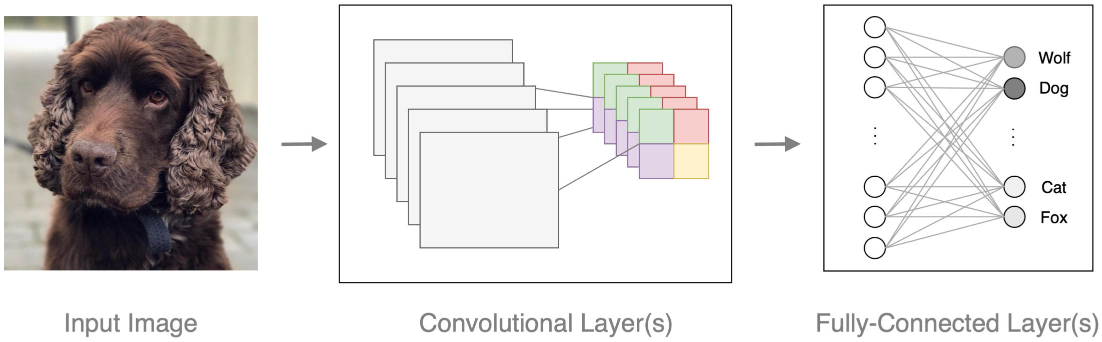
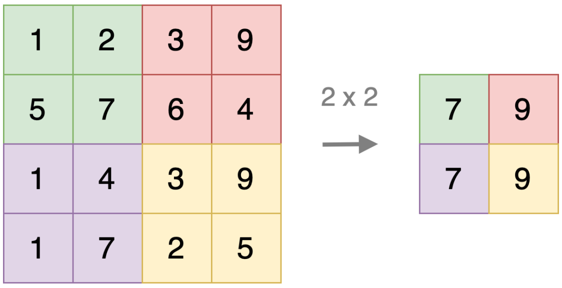


Figure 2.4: Simplified architecture of a convolutional neural network (CNN). The input image is processed through multiple convolutional and pooling layers. The output, after being flattened, is passed to fully-connected layers resulting in softmax probabilities. The darkest neuron corresponds to the class with the highest probability, in this case, a dog.

CNNs have been applied in a variety of fields, such as image and video recognition, recommender systems, image generation, and natural language processing, achieving state-of-the-art results in many of these areas [20].

Max Pooling Max pooling is a subsampling method that selects the maximum value from each of a series of sub-regions of the input, as shown in Figure 2.5. This provides robustness to spatial translations and helps to reduce the computational load for subsequent layers.



1	2	3	9
5	7	6	4
1	4	3	9
1	7	2	5

7	9
7	9

Figure 2.5: Example of max pooling with a 2x2 pooling size. The resulting matrix consists of the maximum values from each 2x2 submatrix in the original matrix.

Average Pooling Average pooling is another subsampling method that calculates the average value for each of a series of sub-regions of the input, as depicted in Figure 2.6. While less common than max pooling, it can be used in scenarios where it is important to maintain the average intensity information of the features.

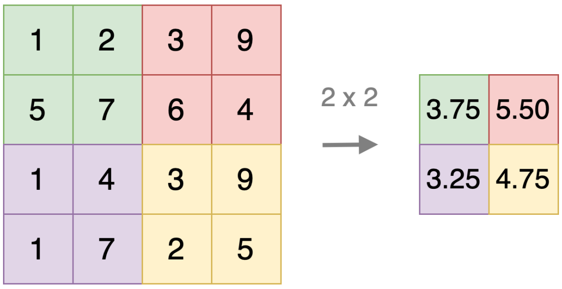


Figure 2.6: Example of average pooling with a 2x2 pooling size. The resulting matrix consists of the average values from each 2x2 submatrix in the original matrix.

Empirical studies have shown that max pooling generally outperforms average pooling in tasks such as object recognition, though the choice of pooling operation can depend on the specific task and dataset [29].

2.3.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural networks that introduce connections between hidden units along the time dimension, enabling the model to maintain information across time steps. This makes RNNs particularly suitable for sequence-to-sequence tasks such as natural language processing, time series prediction, and music generation [30].

Figure 2.7 presents the architecture of a traditional RNN, both in its unrolled and rolled forms. Here, h_t denotes the hidden states and x_t refers to each element of the input sequence.

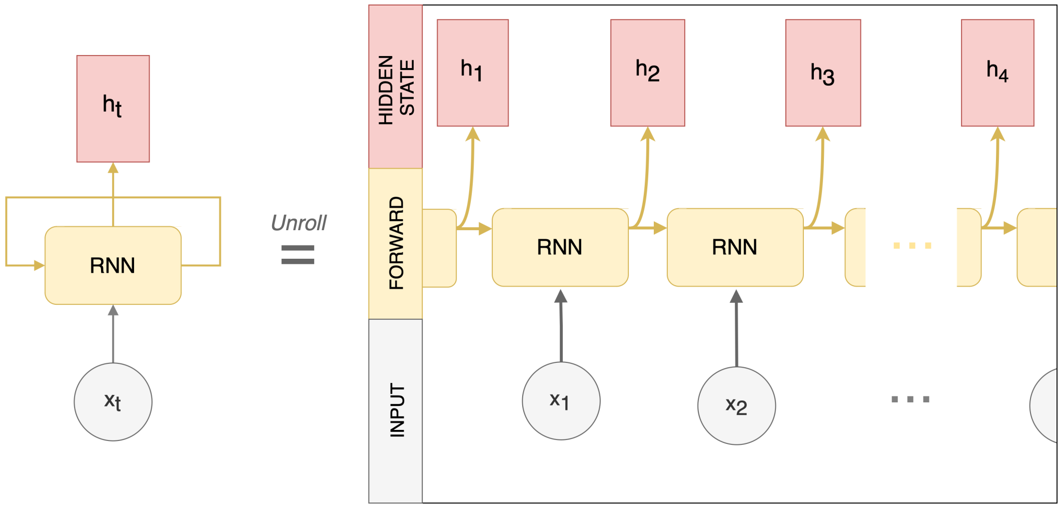


Figure 2.7: Architecture of a traditional recurrent neural network (RNN). h_t denotes the hidden states, while x_t denotes each element of the input sequence. Both unrolled and rolled versions are shown.

Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a special type of RNN designed to overcome the limitation of traditional RNNs in learning long-term dependencies. LSTMs introduce a memory cell and gating mechanisms that regulate the flow of information into and out of the memory cell [30]. This design allows LSTMs to effectively capture long-term dependencies in sequence data.

Despite their effectiveness, LSTMs have a relatively complex structure and require substantial computational resources for training.

Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is another type of RNN that aims to capture long-term dependencies, similar to LSTM. However, GRUs simplify the LSTM architecture by merging the cell state and hidden state, and combining the input and forget gates into a single *update gate* [30]. This results in a more efficient model with fewer parameters than LSTM.

While GRUs simplify the architecture and reduce the training time, it is still a matter of ongoing research to conclusively determine which of the two, GRUs or LSTMs, performs better in various tasks. In some cases, the extra complexity of LSTM may provide an advantage, while in others, the simplicity of GRU may suffice.

Bi-directional Recurrent Neural Network

Bidirectional Recurrent Neural Networks (BRNNs) have been introduced as an improvement over traditional RNNs, which are somewhat limited by their unidirectional processing. BRNNs divide the state neurons into two parts, forward states, and backward states, to process the sequence data in both positive and negative time directions. Importantly, these states are not connected, enabling the BRNN to harness information from both past and future inputs and providing a broader context for making predictions [31]. The network's enhanced modeling and prediction capabilities, facilitated by its ability to capture patterns and dependencies from both time directions, make BRNNs especially valuable for tasks requiring comprehensive context understanding, such as speech recognition, machine translation, and sentiment analysis [31]. Figure 2.8 displays a simple model of a BRNN.

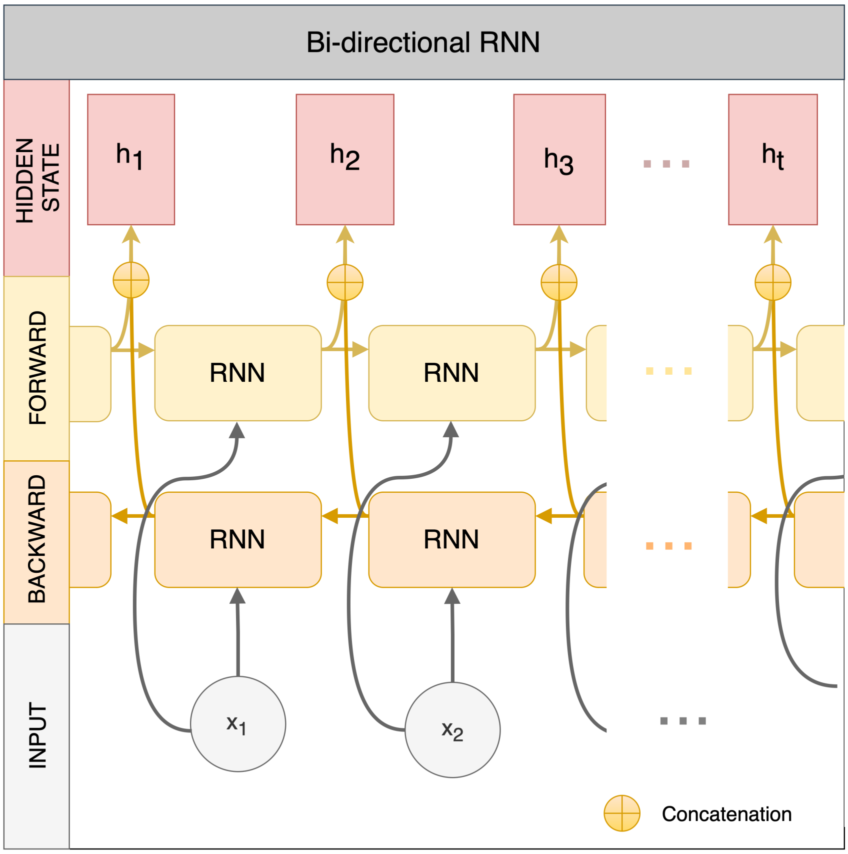


Figure 2.8: Architecture of a Bidirectional Recurrent Neural Network (BRNN). It extends the traditional RNN model by including a second layer of RNN that processes the input sequence in reverse, capturing additional context. The outputs from each direction are then concatenated.

2.4 Attention Mechanisms

William James (1890) described attention as the mind's process of selectively focusing on certain aspects of information while disregarding others. He defined it as *the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought* [32]. This concept has been translated into technology through attention mechanisms, particularly in sequence transduction models, where it allows the model to focus selectively on certain parts of the input sequence that are most relevant for each step of the output sequence.

In the landmark paper Attention Is All You Need, Vaswani et al. introduced the Transformer, a novel network architecture that hinges solely on attention mechanisms, bypassing the need for recurrence or convolutions. The Transformer's significant improvements in both quality and speed over previous models, along with its greater parallelization capabilities, marked a shift in the field of natural language processing and machine translation [33].

2.4.1 Self-Attention

Self-attention, or intra-attention, is a mechanism employed in the Transformer model to capture relationships and dependencies within a sequence. It allows each element in the sequence to attend to every other element, thereby capturing global patterns and long-range dependencies, irrespective of their distance in the sequence [33].

In the self-attention process, each element in the sequence is considered as the query, key, and value. The query and key are used to compute an attention score, which is then used to weigh the value. In essence, the input attends to itself, leading to the term *self-attention* (see Figure 2.9). The resulting output is a weighted sum of the inputs, which considers both the relevance of each input element to every other input element and their original representations.

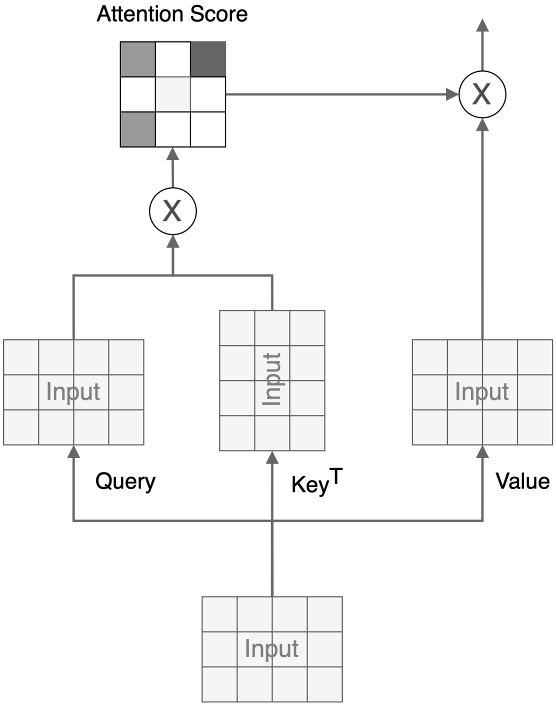


Figure 2.9: Illustration of self-attention. The input sequence is used as query, key, and value. The attention scores are computed using the query and key (transposed), which are then applied to the value through a multiplication operation. The mechanism essentially allows the input to attend to itself. The figure is inspired by the model of Praphul Singh[2].

Self-attention effectively models dependencies within the input sequence, thus leading to superior performance in various sequence transduction tasks such as machine translation [33].

Multi-head Attention

Multi-head attention, an extension of self-attention, is another critical component of the Transformer model. It allows the model to focus on different parts of the input sequence across multiple representation subspaces simultaneously [33].

In multi-head attention, the input is divided into a number of equal parts corresponding to the number of heads. Then, self-attention is applied independently to each part, which allows the model to focus on different positions in the sequence in parallel. This allows the model to capture various types of dependencies and relationships within the sequence. Figure 2.10 provides a visual representation of the multi-head attention mechanism in action, highlighting its unique ability to capture different types of information from the same sequence.

[2] https://blogs.oracle.com/ai-and-datascience/post/multi-head-self-attention-in-nlp

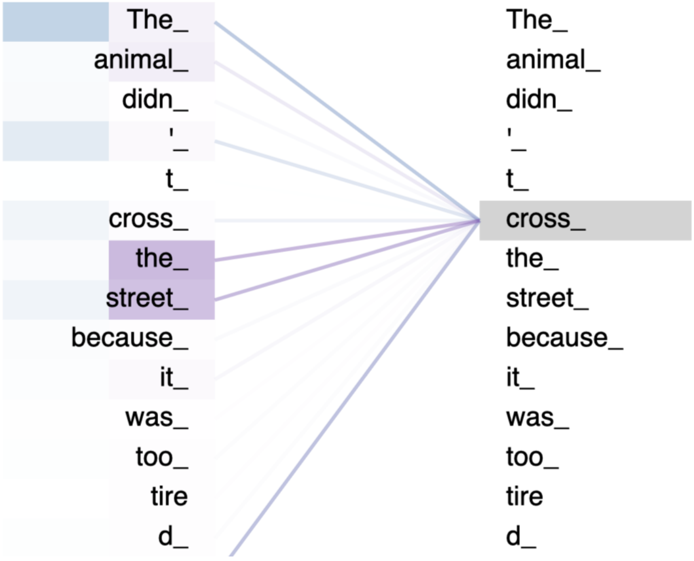


Figure 2.10: Visualization of multi-head attention. In this example, two attention heads (blue and purple) are applied to a sentence. The word *cross* focuses more on *the* and *street* in the first head, and on *The* in the second head. The representation is stronger than if only one head (self-attention) was used. This illustration was created using the Tensor2Tensor Colab notebook[3].

Through this mechanism, the model is capable of capturing diverse relationships across different positions and representational spaces within the sequence, leading to richer understanding and improved performance in tasks such as language modeling and machine translation [33].

Co-Attention

Co-attention is an attention mechanism that considers two modalities concurrently. Unlike traditional self-attention (or intra-attention), which focuses on the relationship among elements within a single modality, co-attention takes into account the interactions between two different modalities simultaneously [34].

In a co-attention mechanism, the representations of the two modalities influence each other's attention processes. More specifically, the attention over one modality is guided by the representation of the other modality, establishing a symmetry in their interactions. This approach is effective in tasks that require an understanding of the relationship between different types of input data.

The process of co-attention is conducted at different levels, enabling the model to capture hierarchical structures within the data. For example, it can be applied at the word, phrase, and sentence levels when dealing with textual data. It can also be performed in

[3] https://colab.research.google.com/github/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/hello_t2t.ipynb#scrollTo=odi2vIMHC3Rm

parallel or alternating fashion, generating attention for both modalities simultaneously or switching between the two.

The primary advantage of co-attention is its ability to capture complex dependencies between different types of input data. It allows the model to focus on the relevant parts of both modalities, leading to better performance in tasks that require an understanding of the relationship between different types of data [34].

2.4.2 Hierarchical Attention Networks

The Hierarchical Attention Network (HAN) is a model developed for tasks that involve structured data, such as document classification. HAN is designed to capture the hierarchical structure of documents by incorporating different layers of attention, focusing on different levels of information within the data [35].

The HAN architecture consists of two primary components: a word sequence encoder with a word-level attention layer, and a sentence encoder with a sentence-level attention layer (Figure 2.11). Both encoders use bidirectional Gated Recurrent Units (GRUs) to capture the contextual information within words and sentences.

In the word sequence encoder, the word-level attention layer applies attention mechanisms to highlight important words within a sentence, based on their contextual information. Similarly, the sentence encoder uses a sentence-level attention layer to select informative sentences within a document, forming a document representation that emphasizes relevant sections.

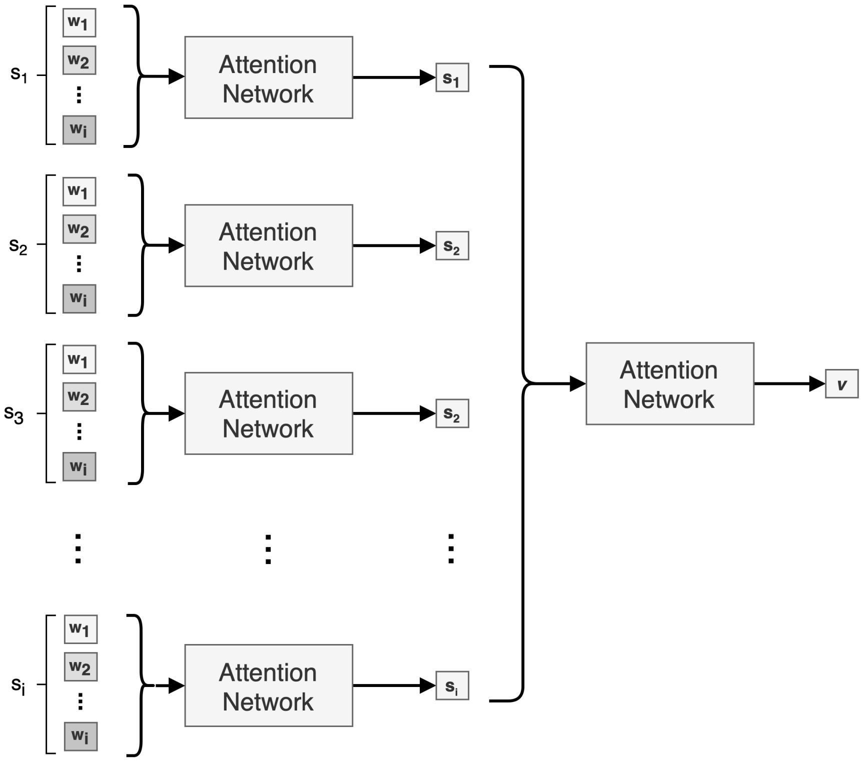


Figure 2.11: Hierarchical Attention Network (HAN) architecture. The network processes each word w_i in a sentence s_i through the attention network, generating a condensed and enhanced sentence representation based on the significant words. These sentence representations are then processed through another attention network to generate an enhanced document representation that emphasizes the important sentences.

A significant advantage of HAN is that its attention mechanisms can reveal the parts of the input data that the model considers most important, providing insights into the decision-making process of the model. This makes HAN not only effective for tasks like document classification but also for the interpretability of text, which is a critical requirement in many real-world applications [35].

The architecture of the attention network within HAN, which includes a bidirectional GRU and an attention score network, is illustrated in Figure 2.12. The network takes an input sequence and generates a series of hidden states using the bidirectional GRU. Each of these hidden states is then fed into the attention score network to generate an attention score. These scores are used to create a weighted sum of the hidden states, which forms the output of the network. This architecture allows the model to give more importance to the significant parts of the input sequence, leading to more effective representations.

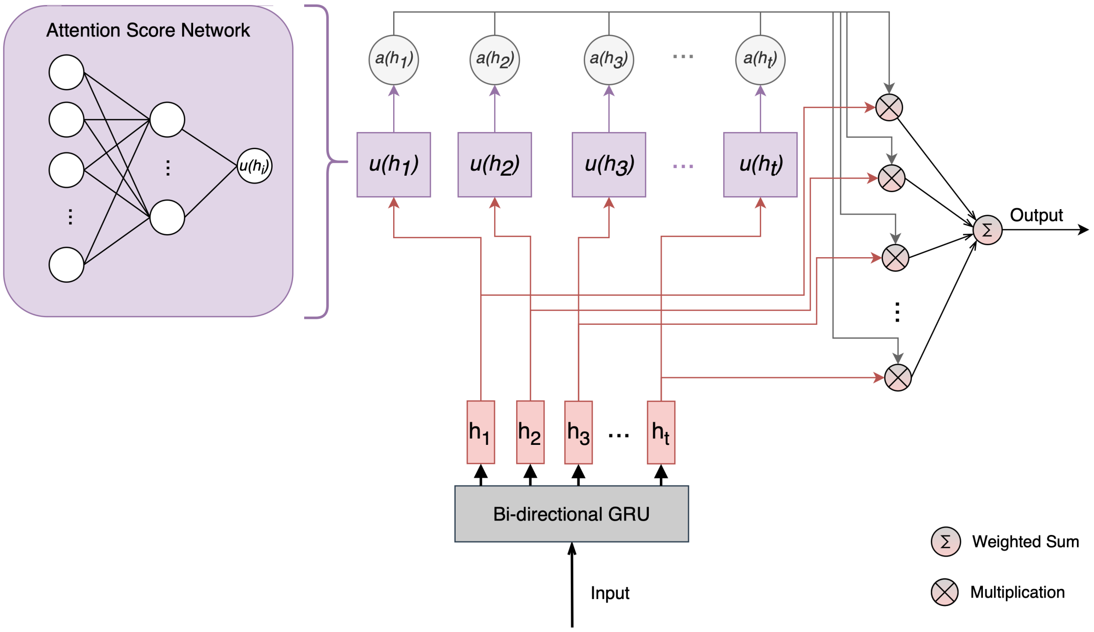


Figure 2.12: Architecture of the attention network within HAN. The network comprises a bidirectional GRU and an attention score network. The input sequence is processed through the bidirectional GRU, generating a series of hidden states. Each hidden state is then fed into the attention score network to generate an attention score, which is used to create a weighted sum of the hidden states that form the network's output. This process allows the network to emphasize important parts of the input sequence.

Overall, the hierarchical attention mechanism introduced by the HAN model allows for a more nuanced and context-aware representation of data at different granularity levels, which is particularly useful for document classification tasks [35].

2.5 Embeddings

Embeddings, according to Jurafsky and Martin [36], represent a potent method in machine learning and deep learning systems to convert raw data into a form that these algorithms can process. Transforming high-dimensional data into a lower-dimensional vector space is essential because many machine learning algorithms and all deep learning algorithms require numerical input. The transformation process maps each unique item to a representative point in the embedding space. This process approximates preserving the relationships and similarities from the original space. In the following sections, the concept will be further explained and contextualized through different embedding models' discussion and exploration.

Embedding techniques fall into two broad categories: uni-modal and multi-modal approaches. Uni-modal approaches involve mapping data from a single modality, such as text or images, into an embedding space. In contrast, multi-modal approaches integrate multiple modalities, such as text and images, into a single embedding space. Both these categories of approaches will be further discussed.

2.5.1 The Distributional Hypothesis

Introduced by Zellig Harris in 1954 [37], the Distributional Hypothesis is a cornerstone in computational linguistics, stating that words with similar distributions have similar meanings. In other words, words that frequently occur in the same environment share semantic features.

Jurafsky and Martin [36] highlight that the hypothesis forms the basis for vector semantics, where word representations, called embeddings, capture the meaning of words in a vector space. In practice, the advent of machine learning and deep learning has enabled us to represent words as high-dimensional vectors in a mathematical space, referred to as the embedding space. These vectors, with their dimensions signifying latent features of the words, serve as the embodiment of the Distributional Hypothesis. The proximity of the vectors can indicate semantic or syntactic similarity.

Figure 2.13 provides a simplified illustration of this vector space, where words describing similar concepts are situated close to each other. The axes represent latent dimensions in the space. Words like *dog* and *wolf* are close, indicating a level of semantic similarity. In contrast, *cat* and *tiger* are distant along the x-axis but close on the y-axis, revealing a different type of relationship. This scenario typifies the Distributional Hypothesis's underpinning, where contextually alike words are considered to have similar meanings.

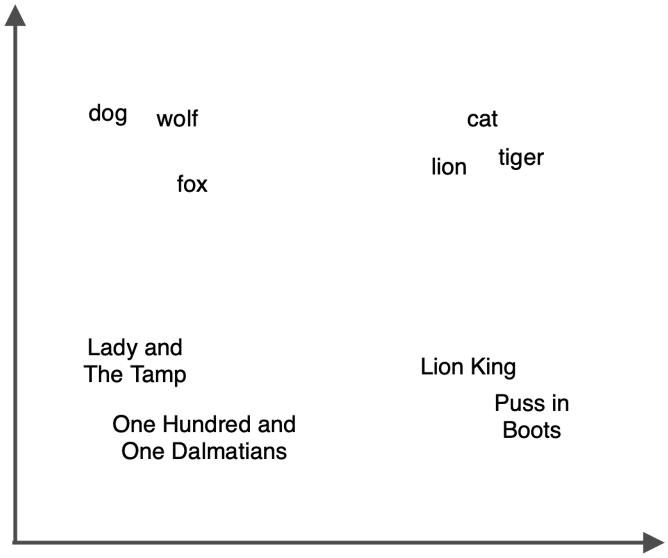


Figure 2.13: A simplified illustration of a vector space under the Distributional Hypothesis. Words are embedded into a multi-dimensional space where the closeness of word vectors represents semantic or syntactic similarity.

The application of the Distributional Hypothesis in machine learning and deep learning has been a stepping stone in the creation of powerful linguistic models, as it provides a method to numerically represent text. This numeric representation forms the basis for word embeddings, which are essential tools for a range of natural language processing tasks.

2.5.2 Word Embeddings

Word embeddings, as discussed by Jurafsky and Martin [36], represent a significant advancement in the way we treat text data in machine learning algorithms. Initially, the most straightforward way of representing text was through Bag of Words models, which involved treating each word as an atomic unit and ignoring the ordering of words [37]. However, this approach did not capture the context or semantics of words. With the advent of more advanced word embeddings such as word2vec, GloVe, and BERT, we now have mechanisms to capture the semantic meanings, syntactic relationships, and contextual associations of words, paving the way for significant improvements in natural language processing tasks. These types of embeddings are discussed in the following sections.

Word2vec

Introduced by Mikolov et al. [38], word2vec is a pioneering method for learning word embeddings. Word2vec aims to learn high-quality word vectors from massive datasets containing billions of words with large vocabularies. These learned word vectors capture multiple degrees of similarity, capturing nuanced semantic and syntactic properties of words.

The core principle of word2vec involves creating dense vector representations of words in such a manner that the vectors for words appearing in similar contexts are closer in the embedding space, embodying the distributional hypothesis [37]. The model utilizes two architectures to achieve this: the Continuous Bag-of-Words (CBOW) and the Skip-gram models. The CBOW model predicts a target word from its surrounding context, whereas the Skip-gram model does the opposite: predicting the context words from the target word. This bidirectional learning allows the model to capture rich contextual information.

This method has led to significant improvements in tasks involving word similarity and has been crucial in downstream tasks such as sentiment analysis, named entity recognition, and information retrieval.

GloVe

Building on the foundations laid by word2vec, Pennington et al. [39] introduced GloVe, short for Global Vectors, a new model for learning vector representations of words. GloVe merges the benefits of both global matrix factorization methods and local context window methods, providing an efficient and powerful mechanism for learning word embeddings.

Unlike word2vec, which learns from local context, GloVe trains on the global word-word co-occurrence counts, capturing global statistical information across the corpus. The aim is to derive meaningful structure and extract semantic and syntactic regularities from these global statistics. This global co-occurrence count allows GloVe to consider a much larger context and better capture long-range dependencies between words.

In terms of performance, GloVe has shown state-of-the-art results in tasks such as word analogy and named entity recognition, underlining its effectiveness in capturing fine-grained semantic and syntactic regularities in word vectors.

BERT

BERT, short for Bidirectional Encoder Representations from Transformers, is a method introduced by Devlin et al. [40] to learn representations of words by looking at their surrounding context in both directions. Unlike earlier methods such as word2vec and GloVe, which only consider a fixed window around a word, BERT considers all the words in a sentence both to the left and right. This makes it much better at understanding the full context of a word.

BERT is based on the transformer architecture, which we discussed earlier. In essence, transformers use a method called *self-attention* to weigh the importance of words in the sentence. This allows BERT to focus more on words that are important for understanding the context and less on words that are not.

At its core, BERT works by randomly hiding some words in a sentence, known as *masking*, and then trying to predict these masked words from the other words in the sentence. This forces BERT to learn to understand the context and meaning of words.

BERT also learns to understand the relationship between sentences by guessing if one sentence logically follows another.

What makes BERT special is its flexibility. After learning from a large amount of text data in this way, a process known as *pre-training*, BERT can be easily adapted to a wide range of tasks, such as answering questions or identifying the sentiment of a text, with only some additional training. This ability to handle various tasks by simply adjusting its inputs and outputs, coupled with its impressive performance, is what made BERT a breakthrough for word embedding.

2.5.3 Image Embeddings

Images can contain a great amount of information, from colors and shapes to patterns and objects. But machines do not understand images like we do. For a machine to interpret an image, it needs to be converted into a numerical form, known as image embedding.

Unlike text, images contain spatial and hierarchical patterns that can be quite complex. Hence, finding an effective way to represent images numerically is more challenging than with text. Various methods have been developed to tackle this challenge, including Convolutional Neural Networks (CNNs) and newer techniques like Vision Transformers (ViTs). In the following sections, we will explore some popular image embedding methods and discuss their unique strengths and features.

VGG19

VGG19 is a type of Convolutional Neural Network (CNN) developed by Karen Simonyan and Andrew Zisserman [41]. It is called VGG19 because it has a total of 19 layers - 16 of them are for feature extraction and 3 are fully-connected layers for classifying the extracted features. This architecture is illustrated in Figure 2.14.

VGG19 operates on color images of size 224x224 pixels. It first applies a series of filters to the input image, progressively reducing the image size while increasing the number of channels (or depth). Each filter is applied to a small 3x3 region of the image, sliding across the entire image to capture different features. This makes VGG19 especially good at spotting patterns, like lines and curves, that appear in different parts of the image.

There are other versions of VGG, like VGG16, which work similarly but have fewer layers. These architectures are known for their simplicity and effectiveness in tasks like image classification.

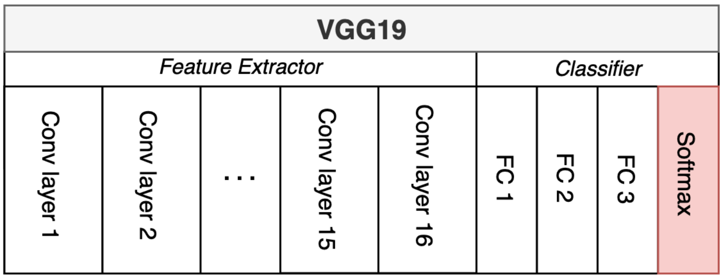


Figure 2.14: VGG19 architecture. This CNN has 19 layers, including a 16-layer feature extractor and a 3-layer classifier. The final layer uses softmax activation for image classification.

ResNet

ResNet, short for Residual Network, is another type of CNN that is designed to be very deep while avoiding the common problem of vanishing gradients, which can make training deep networks difficult [42].

ResNet uses something called *shortcut connections*, which allows the input of a layer to skip one or more layers and be added directly to the output. This helps to keep the gradient from shrinking too much during backpropagation, which makes it easier to train very deep networks.

Like VGG19, ResNet operates on 224x224 pixel images and includes a series of layers for feature extraction and classification. However, the architecture of ResNet is more complex, with different versions having anywhere from 50 to over 100 layers, as shown in Figure 2.15. Despite the additional complexity, ResNet is still efficient and effective for tasks like image classification.

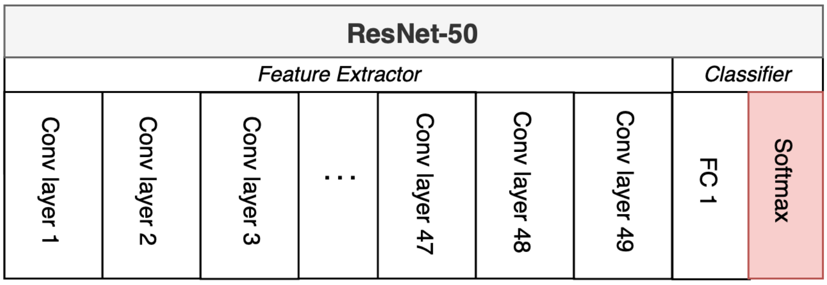


Figure 2.15: ResNet-50 architecture. This CNN has 50 layers, including a 49-layer feature extractor and a 1-layer classifier. The final layer uses softmax activation for image classification.

Inception

The Inception network, also known as GoogLeNet, was first introduced by Szegedy et al. in the paper Going Deeper with Convolutions [43]. It was designed to achieve high performance while efficiently using computational resources. Inception's unique approach relies on approximating an optimal sparse structure with available dense components, allowing it to capture more complex features at different scales.

Inception operates using modules, each consisting of parallel convolutional layers with varying filter sizes (1x1, 3x3, 5x5) and a max pooling layer. This parallel structure helps the model capture features at multiple scales and combine them effectively. Additionally, Inception uses dimension reduction techniques, such as 1x1 convolution layers, to manage computational complexity. This approach prevents computational explosion while preserving the network's capability to recognize complex patterns.

The GoogLeNet variant of the Inception architecture is a 22-layer deep network that uses auxiliary classifiers connected to its intermediate layers. These classifiers help address the vanishing gradient problem and provide regularization during training, although they are discarded during inference. Another feature of GoogLeNet is the use of average pooling instead of fully-connected layers before the final classifier, along with dropout for further regularization.

Inception's design demonstrates an important balance in deep learning: it achieves strong performance on tasks like image classification while remaining computationally efficient. This allows it to function effectively on devices with limited resources, making it more practical for a variety of applications.

Vision Transformers

In contrast to CNN-based architectures like VGG, ResNet, and Inception, Vision Transformers (ViTs) introduced a novel way to address image analysis tasks. Presented by Wu et al. [44], ViTs build on the Transformer architecture, initially developed for natural language processing, and are adapted for image analysis tasks.

The ViT architecture consists of three main components: the tokenizer, the transformer, and the projector. The tokenizer aggregates pixels into semantic tokens, the transformer captures interactions between these tokens, and the projector fuses the transformer's output with the original image features for refined pixel-level details.

ViTs bring several advantages to the table. They focus computational resources more on important image regions, model only relevant concepts, and effectively relate distant but related concepts. Moreover, they can be integrated into existing vision models, replacing parts of a CNN with transformer modules, often resulting in higher accuracy and the need for fewer computational resources.

In essence, ViTs represent a promising alternative to traditional CNNs. They effectively tackle some of the limitations of CNNs, and despite their relative novelty, they have already shown impressive results in various computer vision tasks.

2.5.4 Multi-modal Embeddings

Multi-modal embeddings present an intriguing area in machine learning. They aim to integrate diverse data modalities, such as text, images, and audio, into a shared vector space. This integration facilitates meaningful interaction and comparison between these different modalities, equipping models to tap into the unique insights each modality offers. Prime instances of such a methodology include CLIP and ImageBind, elaborated further in the following sections.

One of the main benefits of multi-modal embeddings is their ability to enable attention mechanisms and other model components to operate seamlessly across modalities. This may result in more accurate and versatile models. Additionally, they could pave the way for zero-shot learning across a range of data types, heralding an exciting phase in machine learning research.

Contrastive Language-Image Pre-Training

CLIP (Contrastive Language-Image Pre-training) is a model that takes a significant step towards multi-modal embeddings by aligning text and images in a shared vector space [45]. As Figure 2.16 illustrates, CLIP is trained by predicting which captions correspond to a given image from a large dataset of image-text pairs, learning to embed related images and text near each other in the vector space.

The strength of CLIP lies in its ability to transfer learning from its pre-training task to a variety of downstream tasks in a zero-shot manner. For instance, it can understand and reference visual concepts learned during pre-training when presented with related natural language, without any further task-specific training. CLIP's performance on various computer vision tasks, such as object classification and action recognition, often matches or surpasses fully supervised models, demonstrating the potential of this approach.

By harnessing the abundance of image-text pairs available on the internet, CLIP underscores the power of natural language as a form of supervision for learning visual representations. This approach is not only scalable but also allows flexible zero-shot transfer by connecting image and text representations, highlighting the promise of multimodal embeddings.

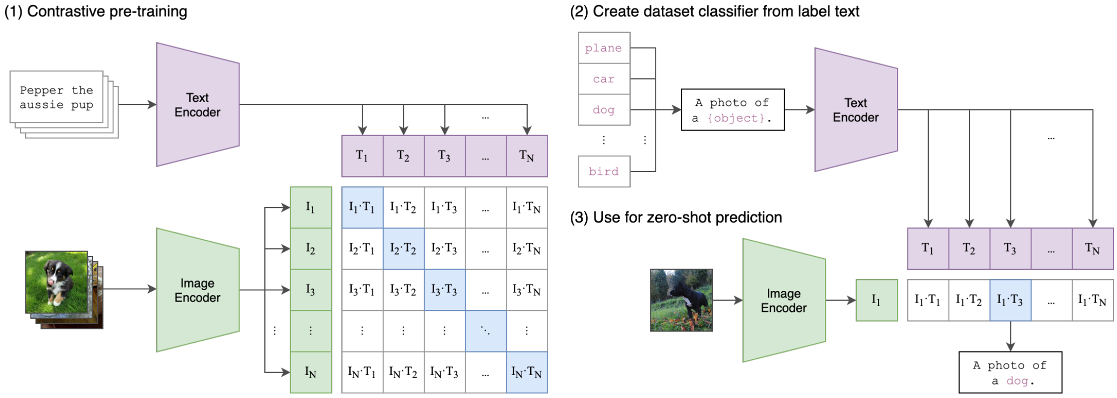

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T1 T2 T3 ... TN
I1 I2 I3 ⋮ IN
I1·T1 I1·T2 I1·T3 ... I1·TN
I2·T1 I2·T2 I2·T3 ... I2·TN
I3·T1 I3·T2 I3·T3 ... I3·TN
IN·T1 IN·T2 IN·T3 ... IN·TN
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I1
I1·T1 I1·T2 I1·T3 ... I1·TN
A photo of a dog.

Figure 2.16: The CLIP model [45] uses a dataset of image-text pairs to align text and images in a shared embedding space. The model can be adapted for zero-shot prediction on various tasks.

ImageBind

Building on the concept of multi-modal embeddings introduced by CLIP, ImageBind takes this one step further by incorporating more types of data modalities into the shared vector space. This includes not only text and images but also modalities such as audio and depth [46].

ImageBind leverages the binding property of images, which is that images can evoke various sensory experiences such as sounds or textures. By aligning the embeddings of each modality to image embeddings, an emergent alignment across all modalities can be achieved. This leads to powerful capabilities, including zero-shot recognition across new modalities and various cross-modal tasks, such as retrieval and detection, and even audio-to-image generation.

ImageBind's approach hints at the future direction of multi-modal embeddings, with increasingly diverse types of data being incorporated into shared vector spaces.

2.6 Fake News Detection

Fake news detection has emerged as a significant research field with the rise of digital media platforms and the subsequent proliferation of misleading information. Primarily, fake news detection is approached as a classification problem, with the majority of algorithms viewing it as a binary classification exercise [2]. This classification formulates the problem into two classes: fake and real news, where the function $Detect(n)$ assigns the value of 1 for real news and 0 for fake news. In mathematical terms, for a given piece of news n,

$$Detect(n) = \begin{cases} 1 & \text{if } n \text{ is real} \\ 0 & \text{if } n \text{ is fake} \end{cases} \tag{2.1}$$

It is, however, worth noting that certain subtypes of fake news classification exist which address more nuanced categories such as satire, humorous fakes, and varying degrees of false information [8].

Machine learning and deep learning approaches have been instrumental in addressing this classification problem. These techniques, including Support Vector Machines (SVMs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs), have been extensively applied and demonstrated promising results [2]. Additionally, data mining techniques and knowledge bases have been leveraged to aid detection [8].

Fake news detection methodologies can be broadly categorized into content-oriented and context-oriented approaches, with hybrid models existing that exploit both types of features [2]. The following sections elaborate on the content- and context-oriented detection strategies.

2.6.1 Content-oriented

The content-oriented approach to fake news detection primarily focuses on extracting and analyzing features from the news content itself. Two types of features are central to this approach: linguistic and visual features [8].

Linguistic features seek to capture different writing styles, sensational headlines, and deceptive cues that are often present in fake news articles. These can be extracted at various text levels, including characters, words, sentences, and documents. They may include lexical features (e.g., word count, unique words), syntactic features (e.g., function words, part-of-speech tagging), and domain-specific linguistic features such as quoted words or external links [8].

Visual features, on the other hand, aim to analyze the images and videos associated with the news articles. Fake news often utilizes sensational or manipulated visual elements to elicit emotional responses from readers. These features could include image ratios, the count of images, and other hand-crafted features at the user and tweet levels [8].

Once these features are extracted, they can be input into various models for the actual task of classification. Two significant types of models in this context are knowledge-based and style-based models. Knowledge-based models rely on external sources for fact-checking claims made in news articles, which could involve expert-oriented, crowdsourcing-oriented, or computational-oriented fact-checking. On the other hand, style-based models exploit the cues found in the writing style of the news content [8].

Expert-oriented fact-checking involves human domain experts investigating the data and documents to verify claims, whereas crowdsourcing-oriented fact-checking aggregates annotations from regular users to assess news veracity. Computational-oriented fact-checking, on the other hand, automates the fact-checking process using external resources like the open web or structured knowledge graphs [8].

2.6.2 Context-oriented

Context-oriented methods for fake news detection leverage social context features, which provide insights beyond the content of the news article itself. These features are generally grouped into three types: user-based features, post-based features, and network-based features [8].

User-based features seek to characterize the users who interact with the news, capturing aspects like registration age, number of followers and followees, and the number of authored tweets. These features can help identify accounts like social bots or cyborgs, which are often associated with the spread of fake news.

Post-based features, on the other hand, analyze the content of the social media posts related to the news article. This includes the user's stance or opinion on the topic, the credibility of the post, and the temporal pattern of the post features over time.

Network-based features focus on the relationships and interactions between users and posts on social media. This includes stance networks, which capture the similarity of stances between tweets; co-occurrence networks, which indicate user engagements with the same news articles; and diffusion networks, which track the spread of news through the platform [8].

Models that incorporate these context-oriented features can be classified into two categories: stance-based models and propagation-based models. Stance-based models infer the veracity of news articles from the viewpoints expressed in related social media posts, using either explicit reactions like *likes* and *dislikes* or implicit sentiment extracted from the post's content. Propagation-based models, on the other hand, predict the credibility of news articles by analyzing the interrelations and spread of related social media posts [8].

2.6.3 Evaluation Metrics

During the evaluation of fake news detection algorithms, a variety of metrics are employed, drawing upon the components of the confusion matrix depicted in Figure 2.17. These components encompass true negatives (TN), false negatives (FN), true positives (TP), and false positives (FP) [8]. TP represents instances where the algorithm correctly identifies news as genuine, while FP denotes the algorithm's incorrect classification of fake news as real news. FN arises when the algorithm mistakenly labels real news as fake, whereas TN corresponds to the algorithm's accurate identification of fake news articles as fake.

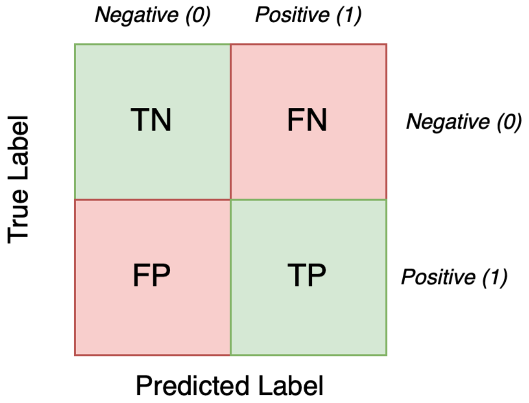


Figure 2.17: A confusion matrix showing the four components of binary classification: True Negatives (TN), False Negatives (FN), True Positives (TP), and False Positives (FP). True predictions are highlighted in green and false predictions are in red.

Based on these components, the following metrics are commonly used for evaluating the performance of fake news detection algorithms:

Accuracy Measures the overall correctness of the classifier, including both true positive and true negative predictions [8]. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

Precision Measures the fraction of true positive predictions among all positive predictions. In the context of fake news detection, precision helps identify which articles are correctly predicted as fake [8]. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

Recall Measures the fraction of true positive predictions among all actual positive instances. In fake news detection, recall quantifies the ability of the classifier to detect all fake news articles [8]. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

F1 Score A single metric that combines both precision and recall using the harmonic mean, thereby giving a balanced measure of the classifier's performance [47]. It is defined as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.5}$$

Micro F1 In multiclass settings, Micro F1 calculates precision and recall for each instance and then averages them [47]. It is particularly useful when the dataset is imbalanced.

Macro F1 Also useful in multiclass settings, Macro F1 calculates precision and recall for each class separately and then averages them [47]. It gives equal weight to each class, regardless of its size.

These metrics allow the comprehensive evaluation of a fake news detection classifier, considering various aspects like precision, recall, and overall accuracy. An additional tool often used is the Receiver Operating Characteristics (ROC) curve and its associated Area Under the Curve (AUC), which provides a measure of the classifier's ability to rank fake news higher than genuine news, considering the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) [8, 47].

2.7 Knowledge Extraction

Knowledge extraction refers to the process of acquiring relevant information from unstructured natural language documents and representing them in a structured format [48]. The primary aim is to transform unstructured knowledge into structured representations that machines can understand and utilize, bridging the gap between human language and machine-readable data. This automated extraction and structuring of knowledge enable various applications, including decision support systems, question answering, and search optimization.

A notable application of knowledge extraction is in the detection of fake news, where it has been applied to enrich the representation of news articles and leverage the relationships among news entities [49, 50]. For instance, the Knowledge-aware Attention Network (KAN) model incorporates external knowledge from a knowledge graph to enhance fake news detection [49]. On the other hand, the CompareNet model utilizes external knowledge from a knowledge base to determine the trustworthiness of a news document [50]. These techniques show the potential of knowledge extraction in leveraging structured knowledge to support complex tasks like fake news detection.

2.7.1 Graph Theory

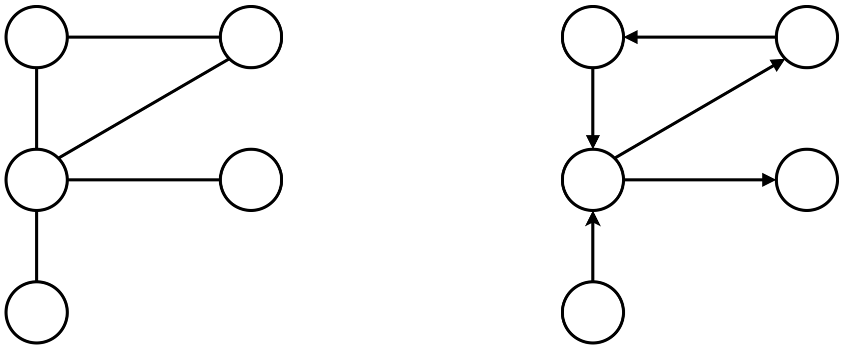
Graph theory is a fundamental area of study that focuses on graphs, and mathematical structures used to model pairwise relations between objects. In the most general sense, a graph is composed of vertices (or nodes) and edges (or arcs). Vertices represent objects or entities, while edges signify connections or relationships between these entities. The degree of a vertex, which is the number of edges connected to it, represents the number of interactions an entity has within the network [51].

The notation used to represent graphs typically involves vertices as points and edges as lines connecting these vertices. Importantly, the specific spatial arrangement or layout of the graph does not influence its properties, as long as the relationships between vertices and edges remain unchanged. The graph depicted in Figure 2.18a is an example of a simple undirected graph. It has five vertices and five edges, representing a simple network where each node is connected to another through an undirected edge. This figure represents a basic structure that can be used to model any pairwise relationship between entities.

Directed Graphs

Directed graphs, also known as digraphs, are a type of graph where the edges have a specific direction, indicating the relationships are not symmetrical but have a certain order [51]. In these graphs, the edge from vertex A to vertex B does not imply a mutual edge from vertex B to vertex A. This characteristic makes directed graphs suitable for representing one-way relationships, such as dependencies or causal relationships.

The directed graph depicted in Figure 2.18b provides an illustration of this concept. Each arrow represents a directed edge, indicating the relationship's direction. It is im-



(a) Undirected graph

(b) Directed graph

Figure 2.18: Illustration of a simple undirected and directed graph with five vertices each connected by either undirected or directed edges, representing two different networks of relationships.

portant to note that the presence of an edge from one vertex to another does not necessitate a corresponding edge in the opposite direction.

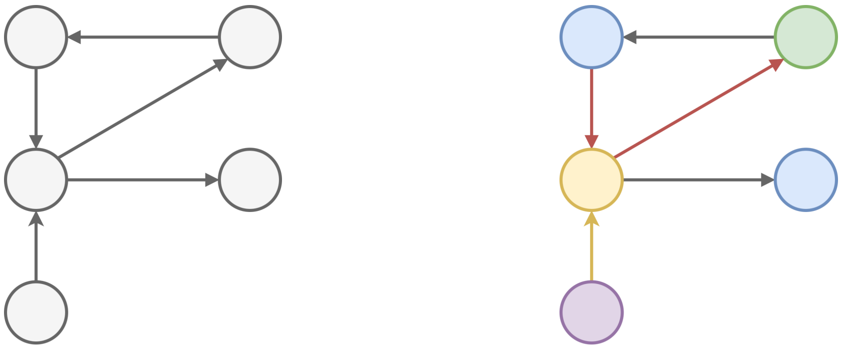
Moreover, the study of directed graphs involves several other concepts, including walks, paths, and cycles [51]. A walk is a sequence of vertices where each adjacent pair is connected by an edge. A path is a walk in which all vertices (and, in the case of directed graphs, all edges) are distinct. Finally, a cycle is a path that starts and ends at the same vertex.

Heterogeneous Graphs

Heterogeneous graphs are a more complex type of graph that contain different types of nodes and links [52]. This is in contrast to homogeneous graphs, where all nodes and links are of the same type, such as for the graphs presented previously. These differences are illustrated in Figure 2.19. The heterogeneous graph, as shown in Figure 2.19b, contains multiple types of nodes and edges, each represented by different colors, while the homogeneous graph, as shown in Figure 2.19a, only contains one type of node and edge.

In a heterogeneous graph, each type of node and edge may have different properties or attributes. For example, in a house-related heterogeneous graph, as seen in Figure 2.20, the nodes could represent different entities like *house*, *building*, and *villa*, each with their unique set of attributes. Similarly, the edges could represent different types of relationships, such as *is* and *can be.*

The heterogeneity of these graphs results in rich and diverse information representation. However, handling this complex structural information and preserving the diverse feature information simultaneously poses challenges in the field of graph theory.



(a) Homogeneous graph **(b)** Heterogeneous graph

Figure 2.19: Comparison of homogeneous and heterogeneous graphs. The coloring illustrates the different entities and relationships found in heterogeneous graphs.

Knowledge Base

A knowledge base serves as the repository of knowledge in an information system, storing sentences expressed in a knowledge representation language. These sentences, often presented as axioms, are assertions about the world [53]. Two main operations are performed on a knowledge base: TELL, which involves adding new sentences, and ASK, which is used to query the existing knowledge. The process of inference is crucial in a knowledge base, allowing the derivation of new sentences based on the existing ones [53].

An example of a knowledge base is the knowledge graph, which is a structured representation of knowledge that is widely used in research and various industries, especially in the fields of Semantic Web technologies, linked data, data analytics, and cloud computing [54]. As shown in Figure 2.20, a knowledge graph is a type of heterogeneous graph that describes real-world entities and their interrelations. A knowledge graph organizes entities, their properties, and their relationships in a graph structure. It is used to define classes and relations of entities in a schema and allows for interrelating arbitrary entities [54]. Notably, the terms *knowledge graph* and *knowledge base* are often used interchangeably, with the former emphasizing the graph structure and the latter focusing on formal semantics [55].

Other examples of knowledge bases include Freebase, YAGO 4, and Wikidata. These are each described below.

Freebase Freebase is a publicly accessible graph database designed to structure human knowledge. Its uniqueness lies in its ability to combine the scalability of structured databases with the collaborative nature of wikis, which facilitates the accumulation of structured information [56]. Freebase demonstrates the power of knowledge bases in effectively organizing, storing, and accessing structured data.

YAGO 4 YAGO 4 is a knowledge base that combines schema.org's typing and constraints with Wikidata's instance data, distinguishing itself with its logical rigor and reasoning capabilities. It maps Wikidata instances to schema.org classes and applies SHACL constraints, enforcing a structured and consistent approach to knowledge representation [57].

Wikidata Wikidata serves as a structured and multilingual knowledge base designed to manage Wikipedia's factual information. With its centralized structure, Wikidata facilitates the access and utilization of structured data from Wikipedia, demonstrating the power of knowledge bases in integrating data, supporting multiple languages, and enabling advanced analytics [58].

In summary, a knowledge base, whether it be a traditional knowledge base or a knowledge graph, serves as a powerful tool for storing and accessing structured information. Examples like Freebase, YAGO, and Wikidata provide valuable resources for integrating, storing, and analyzing diverse sources of knowledge.

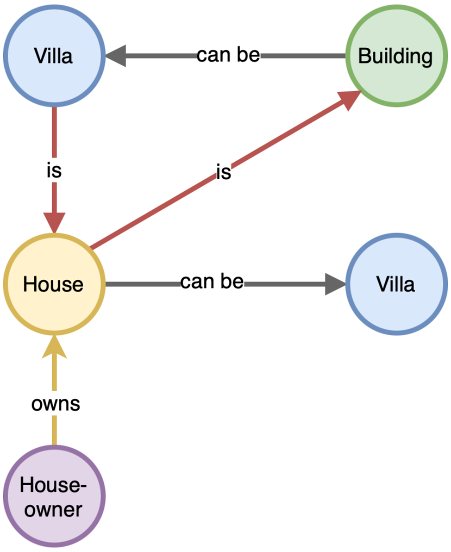
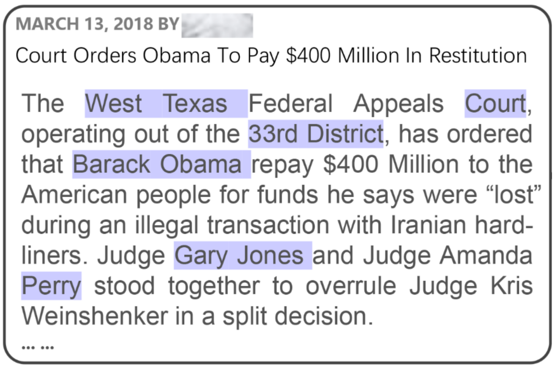


Figure 2.20: Example of a knowledge graph, which is a type of heterogeneous graph. Different relations, like *is* and *can be*, and different vertex types, such as *House* and *Building*, are illustrated.

2.7.2 Entity Extraction, Linking, and Claim Identification

As mentioned in the introduction, knowledge extraction can be viewed as a process of extracting information from a knowledge base. This is a key component of some fake news detection systems, such as the KAN model [49], but the approach can have utility in a variety of other contexts. The entity extraction, linking, and claim identification, the fundamental steps in knowledge extraction, are discussed in detail below.

Entity Extraction The first step in the knowledge extraction process is entity extraction. This process involves identifying and marking the entities present in an arbitrary piece of text, such as that seen in Figure 2.21 [49]. This marking of entities can be accomplished with various tools designed specifically for this purpose. One such tool is the Relation Extraction and Linking (REL) tool[4]. Before the application of such tools, an initial extraction of text from the web can be performed using tools like Newspaper3k [5]. It is important to note that these tools are simply examples of the many available for these purposes.



MARCH 13, 2018 BY [redacted]

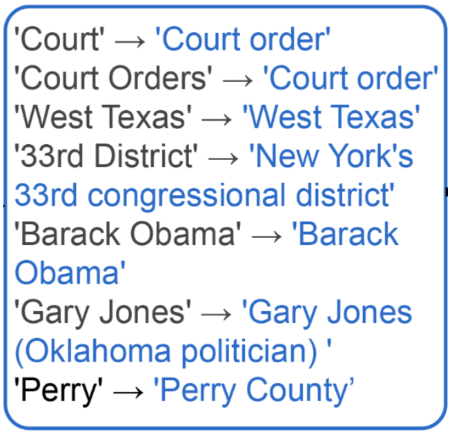
Court Orders Obama To Pay $400 Million In Restitution

The West Texas Federal Appeals Court, operating out of the 33rd District, has ordered that Barack Obama repay $400 Million to the American people for funds he says were "lost" during an illegal transaction with Iranian hard-liners. Judge Gary Jones and Judge Amanda Perry stood together to overrule Judge Kris Weinshenker in a split decision.

... ...

Figure 2.21: An illustration of entity extraction, with entities in an arbitrary text highlighted in purple. Example derived from [49].

Entity Linking Once the entities have been extracted, the next step is entity linking. This process involves mapping each entity extracted from the text to its corresponding linked entities, as demonstrated in Figure 2.22 [49]. Tools such as REL can also be used for this purpose, by identifying mentions of these entities in the text and linking them to their corresponding entities in a knowledge base.



'Court' → 'Court order'
'Court Orders' → 'Court order'
'West Texas' → 'West Texas'
'33rd District' → 'New York's 33rd congressional district'
'Barack Obama' → 'Barack Obama'
'Gary Jones' → 'Gary Jones (Oklahoma politician) '
'Perry' → 'Perry County'

Figure 2.22: An illustration of entity linking, where extracted entities are linked to corresponding entities. Example taken from [49].

[4] https://rel.readthedocs.io/en/latest/

[5] https://newspaper.readthedocs.io/en/latest/

Claim Identification The final step in the knowledge extraction process is claim identification. In this phase, entities that were identified and linked, such as *Barack Obama* in Figure 2.23, are used to search in the knowledge base for associated entities or claims [49]. This step is crucial for the acquisition of structured knowledge that can be further used in various contexts, such as in the detection of fake news. Wikidata, as discussed in a previous section, can serve as a comprehensive knowledge base for such claim extraction.

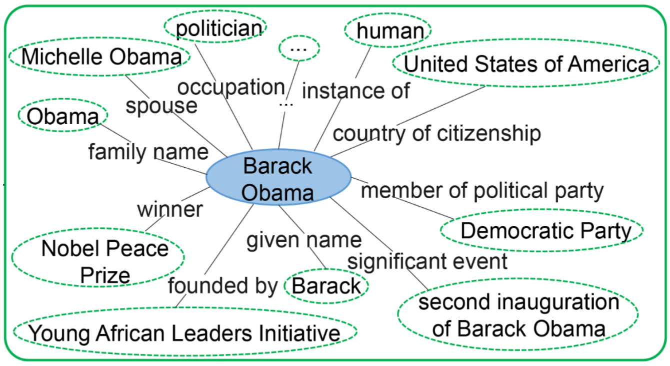


Figure 2.23: An illustration of claim identification, where a given entity is used to search the knowledge base for associated entities or claims. Example derived from [49].

After these steps, the resulting set of claims needs to be embedded for further use in a machine learning setting. This is where tools like Wikipedia2Vec [59] come into play. Wikipedia2Vec provides an efficient and user-friendly solution for learning and visualizing word and entity embeddings from Wikipedia, making it a valuable tool for the embedding process.

CHAPTER
THREE

RELATED WORK

Addressing the complexity and growth of fake news propagation requires advanced and sophisticated detection mechanisms. This chapter explores the evolution from uni-modal approaches, which primarily relies on one attribute of news, such as text, to multi-modal strategies that may integrate various news features such as images. Notably, the Knowledge-aware Attention Network (KAN) [49] and dEFEND [60] are discussed in the context of uni-modal approaches, while the shift towards multi-modal methodologies is represented through models such as FakeMine [10] and the Event Adversarial Neural Network (EANN) [61].

3.1 Uni-modal Fake News Detection

Uni-modal fake news detection models focus on only one aspect of news, typically the textual content. They use a variety of techniques including knowledge graphs, attention mechanisms, and the exploration of the social context surrounding the news, such as comments.

KAN exemplifies the use of external knowledge for enriching the detection process. In this model, entity mentions in the news content are identified and aligned with corresponding entities in a knowledge graph, allowing the model to leverage these entities and their contexts for additional information. This process is known as knowledge extraction, and it was explained in depth in Chapter 2. While promising in its approach, KAN may be limited by its reliance on the quality and comprehensiveness of the employed knowledge graph. It suggests the importance of auxiliary knowledge, but also underlines the need for caution regarding its dependency, as the knowledge of the graph in some cases can be both unreliable and incomplete.

dEFEND [60], on the other hand, prioritizes the integration of explainability into the detection process. It introduces a sentence-comment co-attention sub-network that utilizes both the news content and user comments for its detection. Its primary focus lies in capturing explainable check-worthy sentences and user comments that are vital for fake news detection. However, this approach could face obstacles when dealing with deceptive comments or when user comments are not available for certain news items.

FakeBERT [62] is a deep learning approach that leverages the BERT model, combining it with single-layer deep CNN blocks. FakeBERT effectively handles structured and unstructured text, capturing semantic and long-distance dependencies bidirectionally and offering a more comprehensive analysis of the text content. While this model delivers impressive accuracy, its complexity and computational requirements might pose challenges for larger datasets or real-time applications, implying a trade-off between performance and computational efficiency.

3.1.1 KAHAN

The landscape of uni-modal fake news detection techniques boasts an array of innovative solutions, yet the Knowledge-Aware Hierarchical Attention Network (KAHAN) [9] carves its own niche through a unique approach. KAHAN weaves the external knowledge and social media temporal data into the fabric of uni-modal fake news detection, showcasing their pivotal role in enhancing the performance and effectiveness of the model.

KAHAN is built around dual HANs that simultaneously model the news content and user comments, encapsulating multiple aspects and layers of semantic granularity. To bring an even wider lens to the task, KAHAN incorporates a time-based sub-event division algorithm. This algorithm draws out temporal patterns from user comments, augmenting the detection process by providing a dynamic understanding of user interaction with the news content.

Architecture Overview

As shown in Figure 3.1, the architecture of KAHAN is designed around four major components, namely *external knowledge attention*, *news content encoder*, *user comment encoder*, and the *fake news classifier*.

The *external knowledge attention* module (shown in yellow in Figure 3.1) is responsible for enriching the model's understanding by incorporating external knowledge derived from a knowledge graph. It identifies entities mentioned in the news text, maps these entities to their counterparts in the knowledge graph, and extracts direct neighbors (one-hop neighbors) of the linked entities. Once the entity claims are captured, the entities and their claims are embedded using Wikipedia2vec, creating contextually relevant vector representations. This is the same process that was further elaborated in Chapter 2.

The *news content encoder* (depicted in red in Figure 3.1) uses HAN at the word- and sentence-level to capture the inherent linguistic structure within the news text. Word-level embeddings are aggregated into sentence representations, which are then further combined into an overall news representation using sentence-level embeddings. The News Towards Entities Attention (N-E) mechanism is incorporated to assign importance to entities with respect to the news content. Essentially, this component is a modification of the HAN model discussed in Chapter 2, where entity attention through multi-head attention is used to enhance the sentence representations. This attention-enhanced HAN model is illustrated in Figure 3.2.

The *user comment encoder* (illustrated in blue in Figure 3.1) attempts to model the temporal characteristics of user comments by partitioning them into distinct sub-events. The rationale behind this mechanism is the observation that user perception of a news piece might evolve over time. The Comments Towards Entities Attention (C-E) mechanism is utilized to allocate significance to entities based on their relevance between the entity and the sub-event using external knowledge.

Finally, the *fake news classifier* (shown in green in Figure 3.1) integrates the representations derived from the news content and user comments to determine the veracity of the news. It concatenates the news representation and comment representations into a unified feature vector which is then passed through a fully-connected layer for final predictions.

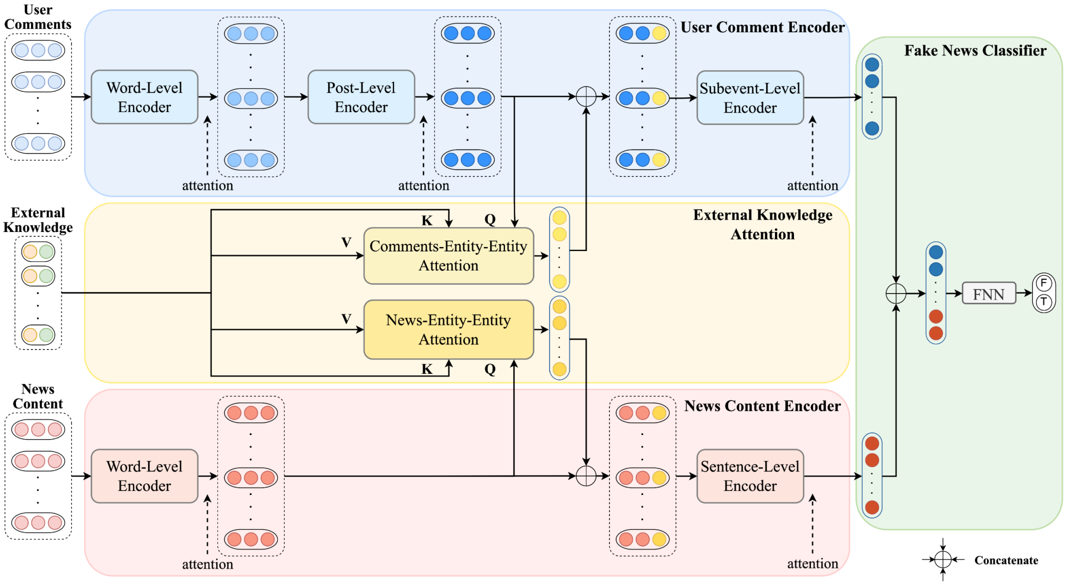


Figure 3.1: The architecture of KAHAN [9]. It consists of four main components: a user comment encoder (blue) that encapsulates the temporal dynamics and thematic structure of user comments, a news content encoder (red) that models the textual information of news content, an external knowledge attention mechanism (yellow) that leverages auxiliary information from a knowledge graph, and a fake news classifier (green) that integrates the learned representations to determine the veracity of the news.

Limitations

KAHAN, along with similar uni-modal fake news detection models such as KAN, exhibit particular constraints that could impede their effectiveness. These limitations primarily

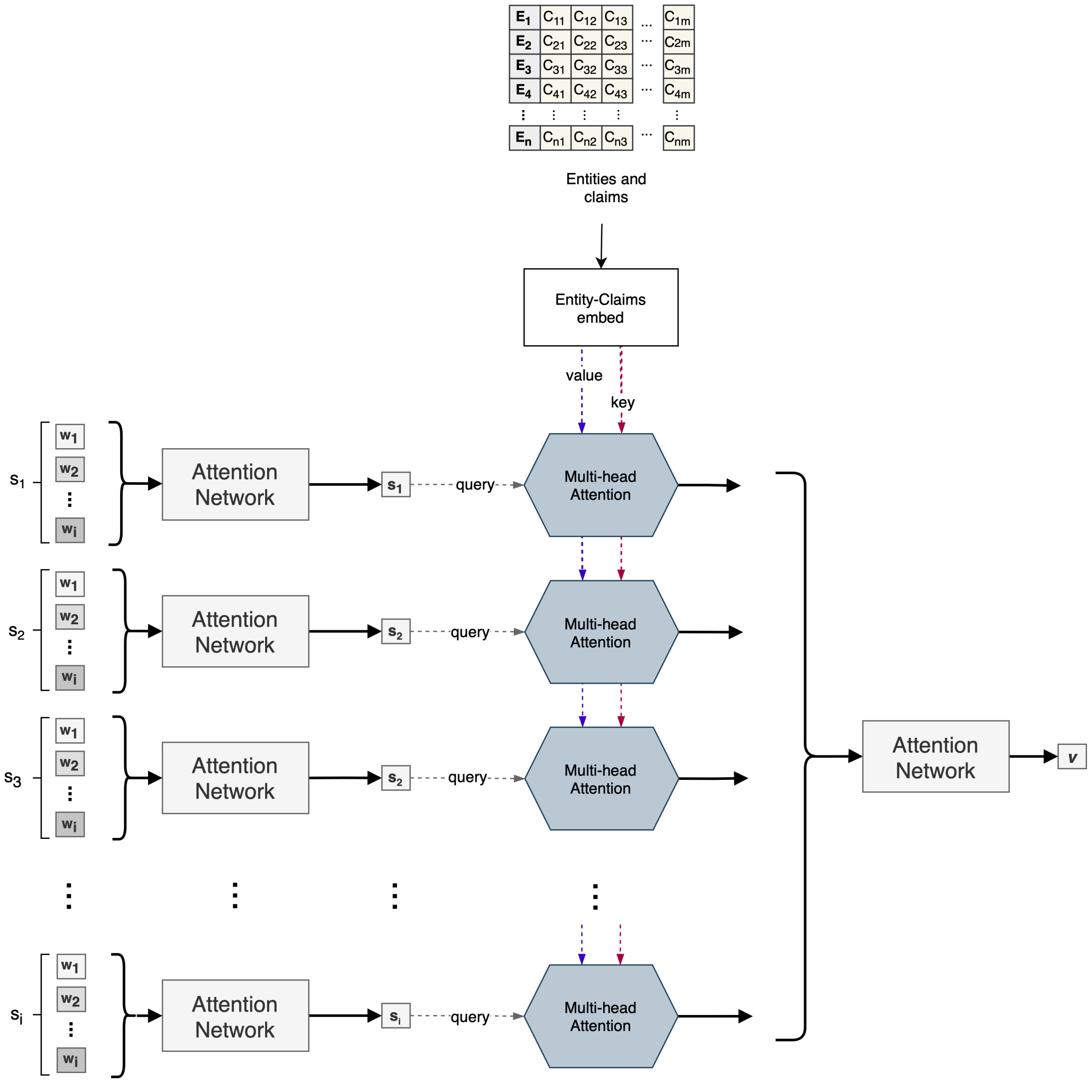


Figure 3.2: Overview of a modification of the HAN architecture that was seen in Figure 2.11. It includes the addition of entity attention through multi-head attention.

revolve around their uni-modal approach, focusing solely on textual data, overlooking the richness offered through other modalities like images.

Firstly, both KAHAN and KAN heavily rely on the quality and completeness of the external knowledge graph. That means if the knowledge graph is inadequate or contains incorrect information, it could negatively influence the models' capability to distinguish between real and fake news. This limitation is not unique to these models but is indeed a common issue with models that rely on knowledge graphs.

Secondly, KAHAN assumes a linear temporal progression of user comments. In the dynamic and often disjointed world of social media interactions, this assumption of linearity may not hold true. User comments can display a non-linear pattern, responding to different aspects of the discussion at various points, which could lead to misinterpretation of the discussion flow, ultimately affecting the accuracy of fake news detection.

However, the possibly most critical limitation of KAHAN, as well as KAN, and the other uni-modal models lies in their dependency on a single modality. This neglects the multi-modal nature of contemporary news, where different types of data such as images, videos, or social network structures often coexist. Cao et al. [12] highlight the importance of a multi-modal approach, emphasizing the pivotal role that for example visual elements can play on detection performance.

This lack of multi-modal consideration might restrict these models' ability to detect fake news that cleverly manipulates multiple modalities. Therefore, the development of a multi-modal approach, integrating various data types alongside text, could significantly enhance the robustness of these fake news detection models.

Drawing upon these insights, it is imperative to extend models like KAHAN and KAN to incorporate a multi-modal approach, thereby making them more versatile and effective in the multifaceted landscape of fake news.

3.2 Multi-modal Fake News Detection

The multi-modal approach to fake news detection is a significant progression from uni-modal methods, leveraging not only textual content but also the wider context of news.

In [10], Ahuja and Kumar introduces FakeMine, a multi-modal model that considers both textual, visual, and network information to detect fake news. This technique uses a Graph Neural Network to explore the network structure of social media posts, BERT to represent textual content while preserving semantic relationships in news articles, and VGG19 to represent image features. An optimized LSTM classifier, enhanced using a Chimp optimization algorithm, is used for final classification. FakeMine surpasses other models in terms of accuracy when tested on multiple modalities, demonstrating the effectiveness of multi-modal fusion and the importance of an optimized classifier in fake news detection. Nevertheless, while FakeMine shows promising results, its complexity may lead to increased computational resources and processing time.

The Event Adversarial Neural Network (EANN) proposed by [61] addresses the challenge of identifying fake news on newly emerged events. EANN is an end-to-end framework that uses an event discriminator to remove event-specific features and maintain shared features among different events. This approach allows the model to learn transferable, event-invariant features, aiding the detection of fake news related to newly arrived events. EANN employs a CNN to extract features from textual and visual content, and its performance has been demonstrated to be superior to existing methods. However, EANN's reliance on adversarial learning could present a limitation if the adversarial components are not carefully regulated, potentially leading to unstable learning dynamics.

Zhou *et al.* presents a Similarity-Aware Fake news detection method (SAFE) [11], which examines multi-modal information, particularly the relationship between textual and visual content in news articles. SAFE uses neural networks to extract textual and visual features and examines their relationship. The representations of news' textual and visual information along with their similarity are jointly learned and used to classify news. This approach aims to identify the incongruity between text and images in news articles, which is often a characteristic of fake news. However, the challenge with SAFE lies in the difficulty of precisely quantifying the semantic similarity between text and images.

Expanding on the concept of multi-modal fake news detection, Zhou et al. [13] introduce the FND-CLIP framework. FND-CLIP leverages the CLIP model and addresses the challenge of cross-modal ambiguity by explicitly calculating the correlation between the text and images in targeted posts. This correlation guides the feature fusion and decision-making stages of the model. Despite its success, FND-CLIP may encounter difficulties when dealing with posts where the text and images exhibit low correlation, potentially leading to less informative fused features.

Lastly, [63] proposes Sentiment-Aware Multi-modal Embedding (SAME), an end-to-end deep embedding framework that incorporates user sentiment into fake news detection. It uses an adversarial mechanism to preserve semantic relevance and representation consistency across different modalities. SAME specifically considers the sentiment hidden in user comments, providing a unique perspective to fake news detection. Despite its innovative approach, SAME might face challenges when dealing with ambiguous or contradictory sentiments expressed in user comments.

The evolution of multi-modal fake news detection, as demonstrated by the works of Ahuja et al. [10], Wang et al. [61], Zhou et al. [11, 13], and Cui et al. [63], has certainly added invaluable dimensions to the field. Each approach, while effective, has its limitations, such as computational complexity, instability from adversarial learning, the precision of semantic similarity quantification, and dealing with low correlation or ambiguous sentiments.

These challenges and the varied strengths of the individual methods highlight the need for a more integrated approach to fake news detection. Such an approach would consider multiple forms of data and techniques, extending the strength of high-performing uni-modal models by incorporating elements of multi-modal methods. The variety of multi-modal methods and their inherent strengths and limitations underscore the necessity of a comprehensive solution. This solution would not only address the limitations of current methodologies but also draw from the salient findings of previous research, thereby enhancing the robustness of fake news detection. This pursuit of an integrated approach and its detailed exploration will be the focus of the following chapters.

CHAPTER

FOUR

DATASETS

In the field of fake news detection, the significance of datasets cannot be overstated, as they serve as the foundation for constructing and validating robust models. This chapter delves into a comprehensive exploration and evaluation of the datasets employed in this study, with a primary emphasis on the collection process, dataset contents, and the associated challenges encountered. These challenges encompass aspects such as diversity and access issues, which significantly impact the quality and reliability of the data. Moreover, this chapter sheds light on the modifications implemented in the collection process to overcome these challenges and enhance the overall data quality.

4.1 FakeNewsNet

In this study, two distinct datasets have been collected and used, namely **PolitiFact** and **GossipCop**, both of which are collected as part of the FakeNewsNet dataset [64]. FakeNewsNet consists of news articles that have been labeled by the fact-checking organizations GossipCop[1] and PolitiFact[2], with GossipCop primarily focusing on news within the entertainment and celebrity domain, while PolitiFact focuses on political and mainstream content. The FakeNewsNet web scraper[3] retrieves not only the news content of the labeled news articles, but also related social media data from Twitter. The FakeNewsNet web crawler consists of multiple parts, first, it gathers true labels from the claims made by fact-checkers, and further, it collects the news content from the URL of the fact-checked article. If the article for various reasons cannot be accessed, a search will be made to WayBack Machine[4] to look for a usable snapshot. Additionally, the crawler utilizes Twitter's search API to find tweets linking to the article, its responses, and details about the relevant users [64].

[1] `https://web.archive.org/web/20200903082521/https://www.gossipcop.com/` (only accessible via the Wayback Machine due to the discontinuation of the organization in 2021).

[2] `https://www.politifact.com`

[3] `https://github.com/KaiDMML/FakeNewsNet`

[4] `https://web.archive.org/`

Due to difficulties gaining access to the Twitter API, the social media data was retrieved from the external data repository[5] collected by the authors of SAFE [11]. In addition, for data quality reasons, the news content was collected using a slight modification of the FakeNewsNet collection process. This will be elaborated further in a later section.

4.1.1 FakeNewsNet Contents and Data Collection

FakeNewsNet offers a rich and diverse set of data features spanning news text, social media data, and images, providing an extensive landscape for fake news analysis. This subsection provides detailed descriptions of the specific data types collected and the process involved in the collection of these data points.

News Text

The news text was collected as JSON files named *news_article.json*. These files contain a set of attributes about the specific article. The utilized attributes of this file that is utilized in this thesis are seen in the list below.

text: The main body text of the news articles.

images: The image URLs associated with news articles, which may include illustrations, photographs, or other visual elements.

top_img: The URL of the most prominent or featured image within the news article.

Social Media Data

The social media data was gathered from a pre-collected data repository and divided into a set of four files, each encapsulating relevant data associated with each news article from Twitter (see the list below).

replies.json: The aggregation of user-generated responses or replies, including information such as the reply text, the original tweet, and the timestamp of the reply.

likes.json: The records of likes attributed to all tweets, detailing the user who liked which tweet.

retweets.json: The data pertaining to retweets, including the original tweet's text, retweet text, and the user who retweeted the content.

tweets.json: The primary data of individual tweets, incorporating tweet metadata such as the tweet's author, creation timestamp, content, hashtags, and other relevant information.

[5] `https://drive.google.com/drive/folders/1gSx4S9i6Haul4TQRkoNQtj3sRHVwGFQ3`

In this thesis, only comments or replies on tweets will be considered, therefore only *replies.json* is relevant.

Images

Images were retrieved mainly using the *top_img* attribute. A Python script was implemented to download the image of every news article. These images were then stored as separate files in a folder structure that distinguished between fake and real news images. In many cases, the script was unable to download specific images because they were missing from the specified URL. In these instances, an addition to the script implemented the Levenshtein distance to find the most similar URL from the *images* attribute list to that of *top_img* and re-attempted the download.

4.1.2 Enhanced Data Collection

As previously indicated, the data collection process for FakeNewNet was refined to enhance the quality of the datasets. This adjustment was necessitated by the substantial volume of irrelevant news text identified in the datasets collected via FakeNewsNet. Table 4.1 presents a few instances of such content, exhibiting the frequency of occurrences in the first column and the specific content in the second. From both datasets, we can infer the crawler's failure in cases where websites display pop-up elements, such as in the fourth row of Table 4.1b. This isn't unexpected given that the crawler lacks the capacity to interact with the browser to remove overlaying elements.

Furthermore, a prevalent issue was discovered, pertaining to *reference decay* or *link rot.* As demonstrated in the first, fifth, and last rows of Table 4.1a, as well as the seventh row of Table 4.1b, the content suggests that the domain is either no longer active or available for sale. Similarly, as shown in the third and penultimate rows of Table 4.1a, the domain persists but the actual news article is absent, seemingly leading the crawler to redirect to the home page and scrape the content there. Additionally, in certain cases, the incorrect section of the website is scraped, such as in the sixth row of Table 4.1b, where the domain owners' policy notice is obtained instead of the actual article text. This highlights the limitations of the Newspaper3k library, which is utilized by the crawler for web article content extraction.

Table 4.1: Top 10 most frequent news content occurrences in the PolitiFact and GossipCop datasets. Instances of irrelevant or redundant content suggest crawler inefficiency and webpage navigation issues.

(a) PolitiFact Dataset

News Content	**Freq.**
Everygame 0.0 rating GET $750 IN BONUS FUNDS ON YOUR FIRST THREE DE...	15
Username Password Need help? Contact the CQ Hotline at(800) 648-2848.	13
About Trendolizer™ Trendolizer™ (patent pending) automatically scans ...	13
JavaScript is not available. We've detected that JavaScript is disa...	9
Yes, you can transfer your domain to any registrar or hosting compa...	7
For full functionality of this site it is necessary to enable JavaS...	6
Use this guide to help you find the full text of recent bills and r...	5
COPYRIGHT © 2005 LexisNexis, a division of Reed Elsevier Inc. All r...	4
The .gov means it's official. Federal government websites often end...	4
The site is unavailable. CQ.com is currently unavailable. We are wo...	4

(b) GossipCop Dataset

News Content	**Freq.**
When you face the world, all you want is to be seen as the unstopp...	95
* Please note that this form cannot be used to reset your Google Ac...	76
A big swirling bucket of the latest rumors, celebrity news and Holl...	62
We use cookies on our website to give you the most relevant experie...	59
You are using an older browser version. Please use a supported vers...	56
IMDb.com, Inc. takes no responsibility for the content or accuracy ...	53
The domain nextdivas.com is for sale. To purchase, call BuyDomains....	51
About Trendolizer™ Trendolizer™ (patent pending) automatically scan...	47
et Google-selskap levere og vedlikeholde Google-tjenester spill, bi...	35
Enter the characters you see below Sorry, we just need to make sure you're not a robot...	31

To counteract the aforementioned issues, two modifications were made to the FakeNewsNet web crawler: (1) prioritizing Wayback machine snapshots of web articles, falling back on the original URL only when snapshots were unavailable, and (2) integrating a randomized user agent on each request to circumvent human verification during high request frequency. The latter issue was observed in the 31 cases outlined in the last row of Table 4.1b. The first modification seeks to solve the *link rot* issue, where URLs no longer point to the intended website, or the website has been shut down or relocated. This refined data collection process will henceforth be referred to as FakeNewsNet+.

Table 4.2: Comparison of the number of news articles, images, and average comments per news between the original FakeNewsNet and the enhanced FakeNewsNet+ crawler. Improvement in data quantity and quality with FakeNewsNet+ is evident. The best numbers are marked in bold.

	FakeNewsNet+	FakeNewsNet
PolitiFact		
# Real News/Images	**624/219**	408/187
# Fake News/Images	**432/172**	351/143
# Total News/Images	**1056/391**	759/330
Avg. # Comments per News	**184**	163
GossipCop		
# Real News/Images	**16817**/1564	13416/**1973**
# Fake News/Images	**5323**/1779	4256/**2033**
# Total News/Images	**22140**/3343	17672/**4006**
Avg. # Comments per News	8	8

Table 4.2 offers a comprehensive comparison between the data gathered by FakeNewsNet and its enhanced version, FakeNewsNet+. A significant increase in the number of fake and real news articles gathered by FakeNewsNet+ stands out. Moreover, FakeNewsNet+ collects more images for the PolitiFact dataset. However, it retrieves fewer images than FakeNewsNet for GossipCop, indicating potential constraints in image retrieval reliability from the WayBack Machine. This is consistent with the WayBack Machine's help page [65] stating that images aren't always archived with webpages. Nonetheless, the images gathered with FakeNewsNet+ are likely more reliable as they're sourced directly from the archive, ensuring their authenticity. Likewise, the news content scraped with FakeNewsNet+ is more likely to originate from the correct source.

Additionally, the average number of comments per news in the PolitiFact dataset has seen a considerable increase. This can be attributed to the additional news articles gathered by FakeNewsNet+, which appear to have more associated comments in the aforementioned data repository compared to the other news articles.

In conclusion, the enhanced version, FakeNewsNet+, provides a significant increase in data quantity along with indications of improved data quality.

4.2 Data Cleaning and Preparation

The process of data cleaning and preparation was conducted diligently on the gathered datasets to augment their quality. Upon an initial evaluation, certain weaknesses were identified in the GossipCop and PolitiFact datasets. Each feature utilized in this study, the **news text**, **comments**, and **image** feature, posed distinct challenges, which are discussed in detail below. A comprehensive overview of the data cleaning and preparation process is available in Table 4.3.

Table 4.3: The table illustrates the progressive removal of cases at each stage, refining the PolitiFact and GossipCop datasets. The final datasets are presented at the bottom, while the accompanying percentages indicate the significance of each step for the size of the datasets.

Stage	Platform	# News	# Images	Avg. # Comments
Original	PolitiFact	1056	417	184
	GossipCop	22140	3409	8
Text Cleaning	PolitiFact	-219	–	+15
	GossipCop	-2535	–	-1
Image Cleaning	PolitiFact	–	-26	–
	GossipCop	–	-66	–
Preparation	PolitiFact	-446	–	+41
	GossipCop	-16262	–	+6
Final Datasets	**PolitiFact**	**391**(-63.0%)	**391**(-6.24%)	**240**(+30.4%)
	GossipCop	**3343**(-84.9%)	**3343**(-1.93%)	**13**(+38.5%)

The refined GossipCop and PolitiFact datasets are publicly accessible on Google Drive[6].

4.2.1 News Text Feature

Enhancements to the dataset collection process aimed to mitigate the inclusion of incorrect or irrelevant data, particularly in the face of changing domains and altered or removed web pages. Despite these measures, some issues persisted in the FakeNewsNet+ datasets, such as the inclusion of duplicate news text. As noted earlier, this problem originates from the weaknesses of Newspaper3k, which were not rectified in FakeNewsNet+. A total of 219 and 2535 news items were eliminated from each respective dataset due to concerns over the text content, as noted in Table 4.3. The cases removed included duplicates, content less than 50 characters long, and non-English content. Short content was deemed suspicious, possibly being unrelated and simply resulting from the aforementioned challenges in data retrieval. Non-English content was excluded to ensure dataset consistency, which is conducive to enhancing classification performance.

[6] `https://drive.google.com/drive/folders/1sIuZ4c3EBzgzShMxM_o6zbtS8GW8dRo6?usp=sharing`

4.2.2 Comments Feature

Similar to the news text, comments that were either non-English or consisted of blank spaces or empty content were removed to ensure consistency. This cleaning process did not directly influence the quantity of data in either dataset, but it did have a notable effect on the average number of comments per news item throughout the cleaning and preparation process (as seen in Table 4.3).

4.2.3 Image Feature

Upon evaluating the datasets, it became evident that some images were not directly related to the corresponding news articles. Such irrelevant images were either discarded or, when feasible, manually replaced with appropriate ones accessed directly from the relevant websites. A set of such images from the datasets are demonstrated in Figure 4.1. An intriguing example includes a pizza advertisement image that seemingly has no link to the news story. Additional issues were identified with certain images, such as monochrome images, loading icons, or images displaying text indicating their unavailability. All such images were removed from the datasets, resulting in the removal of 26 images from the PolitiFact dataset and 66 images from the GossipCop dataset.

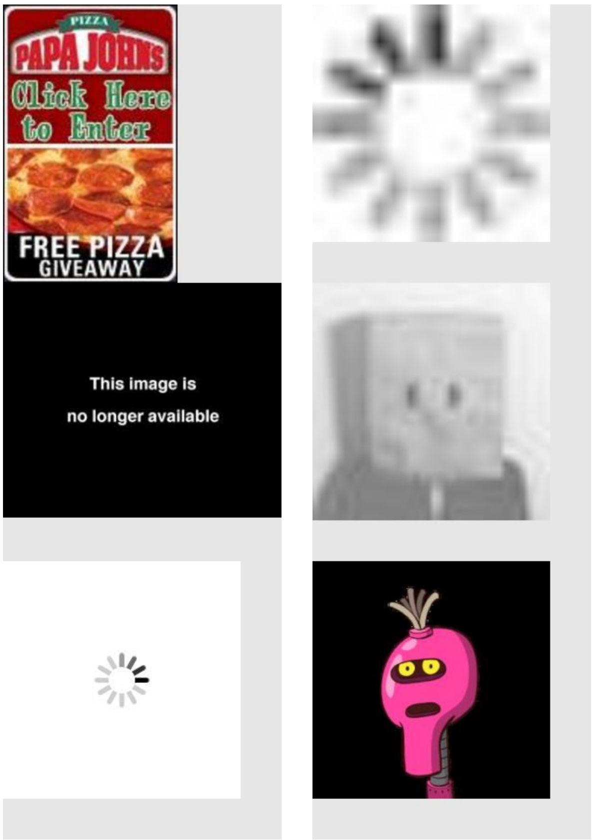


Figure 4.1: Examples of images that were removed from the datasets during cleaning.

An interesting observation in the image dataset was the existence of images that matched those shown in Figure 4.2. These images could potentially leak the labels of the news pieces, causing the model to cheat during training and appear more accurate in classifying news than it would be on real-world data. It is possible that this could happen because the numerical representation of the images might preserve certain characteristics associated with them. For example, if an image with a red color and contained the text *False* consistently appeared alongside certain news articles, the classifier might learn that any such similar image should be labeled as fake news. Since these cases were not very frequent in the dataset, and due to the limited amount of data, such cases were not removed from the dataset. Moreover, given the relatively small number of instances, it is unlikely that a classifier would have sufficient exposure to learn this pattern.

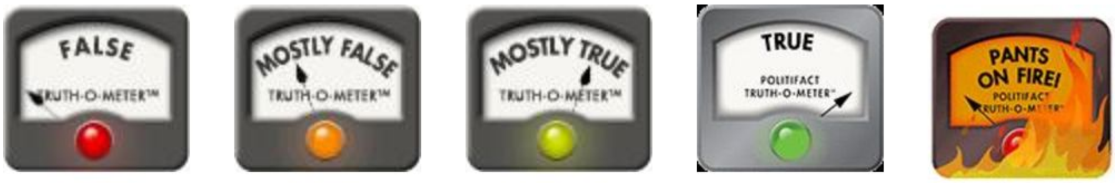


Figure 4.2: Figure depicting the images used by the PolitiFact fact-checkers to indicate the degree of truth of a news piece.

The final stage of our data cleaning and preparation process was data preparation, where we removed all news articles without an associated valid image or any news text. This action led to a significant reduction in the number of news articles in our datasets, potentially limiting the maximum achievable classification performance. However, to maintain accurate and complete data, this trade-off was deemed necessary.

4.3 Data Presentation and Visualization

The final datasets are presented in Table 4.4. The PolitiFact dataset consists of 391 news articles, while the GossipCop dataset contains 3343 news articles. These figures represent a reduction of -63.0% and -84.9% compared to the initial datasets before the preparation and cleaning process, as shown in Table 4.3. Additionally, there is a slight decrease in the number of images and a notable increase in the average number of comments per news article for both datasets.

Table 4.4: Statistics for the PolitiFact and GossipCop datasets after data cleaning.

	PolitiFact	GossipCop
Avg. # Comments per News	240	13
Avg. # Entities per News	105	28
Avg. # Entity Claims per News	37	24
Real News	219	1564
Fake News	172	1779
Total News	**391**	**3343**

An interesting observation from Table 4.4 is the minor imbalance in the distribution of real and fake news within both datasets. The PolitiFact dataset contains more real news articles, while the GossipCop dataset contains a higher proportion of fake news articles. This data imbalance has the potential to contribute to an increased risk of misclassification if not effectively addressed and handled during the classification process. Another interesting observation is that there are typically much more comments and entity mentions in the PolitiFact dataset compared to GossipCop.

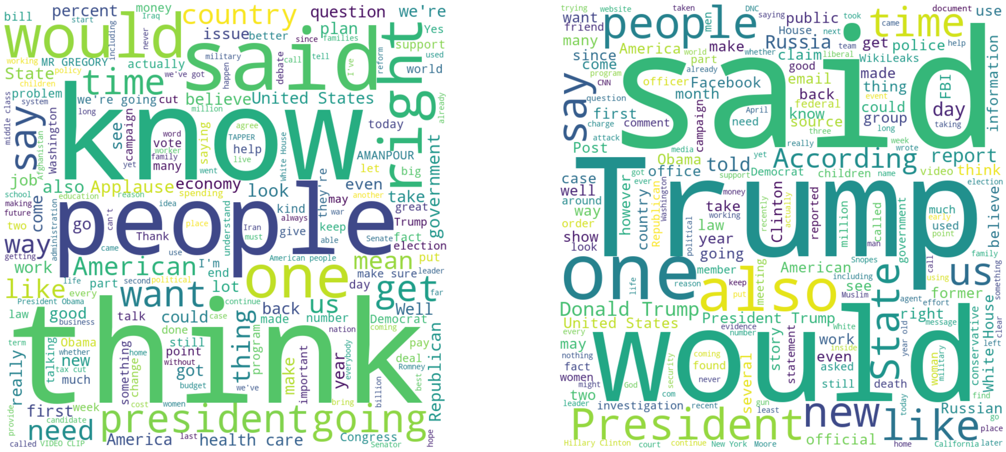
In the upcoming sections, data from each dataset will be visualized and discussed the data, with the primary objective to uncover any discernible distinctions, particularly in relation to differentiating between real and fake news.

4.3.1 PolitiFact

The textual contents of the datasets can be better understood through the analysis of word clouds. The word clouds generated from the PolitiFact dataset, as shown in Figure 4.3, provide insights into the language used in both the fake and real news articles. In the word cloud associated with fake news, the most prominent words are *Trump*, *said*, and *would*. These words suggest a focus on controversial statements and political rhetoric often associated with fake news. On the other hand, the real news word cloud prominently features words such as *think*, *know*, and *people* indicating a more informative and factual tone in the real news articles.

The differences observed in the word clouds between fake and real news in the PolitiFact dataset can be attributed to the typical presentation of fake news. Fake news generally relies on sensationalism and exaggeration to attract attention. As a result, it tends to prioritize provocative language and controversial figures such as former President Trump. In contrast, real news articles appeal to a broader audience, commonly with a more objective and informative tone. The presented word clouds largely confirm these distinctions.

PolitiFact

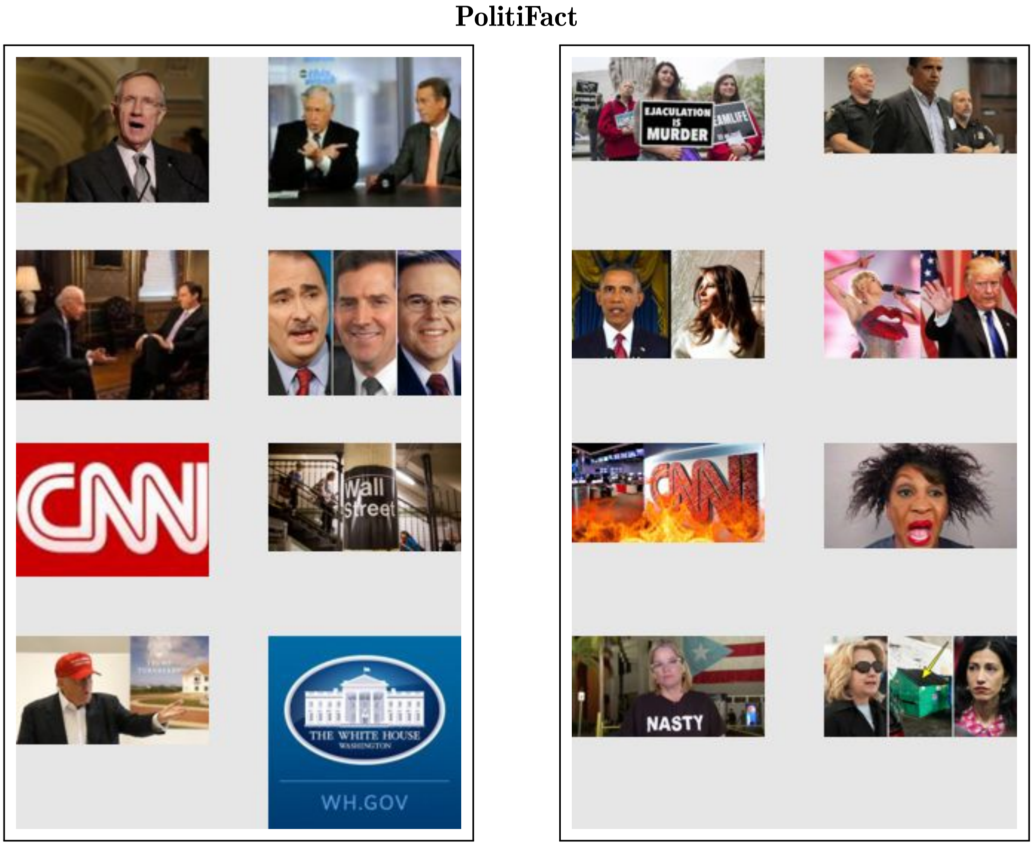


(a) Real News

(b) Fake News

Figure 4.3: Word clouds depicting the most frequent words in the PolitiFact dataset, differentiating between real and fake news articles.

In addition to the textual content, images also play a significant role in understanding the datasets. Figure 4.4 displays the images associated with real and fake news articles in the PolitiFact dataset. The images related to real news articles typically feature politicians, news organizations like CNN, and professional-looking photographs of individuals or locations, as depicted in Figure 4.4a. These images convey a sense of seriousness and professionalism. In contrast, the images accompanying fake news articles exhibit a higher level of extremism and a lack of seriousness. For example, Figure 4.4b shows the CNN logo with flames edited onto it, as well as two images with superimposed text containing provocative statements like *Nasty* and *Ejaculation is murder*. Additionally, fake news images sometimes employ humor, such as a person making a funny face, which can also be observed in the figure. These insights suggest that the visual elements accompanying fake news articles often aim to evoke emotional responses and reinforce sensational narratives rather than providing factual information, aligned with the findings of the discussion on distinguishing attributes of fake news in Chapter 2. In addition, one notable image among the fake news images in Figure 4.4b is the one in the top right corner, where the face of former President Obama has been edited onto another man's body, creating the illusion of an arrest. This manipulated image, along with the cases of superimposed text, further indicates that fake news images are more prone to manipulation and modification. It suggests that these images are intentionally altered to mislead and provoke emotional reactions rather than presenting authentic visual representations.



(a) Real News

(b) Fake News

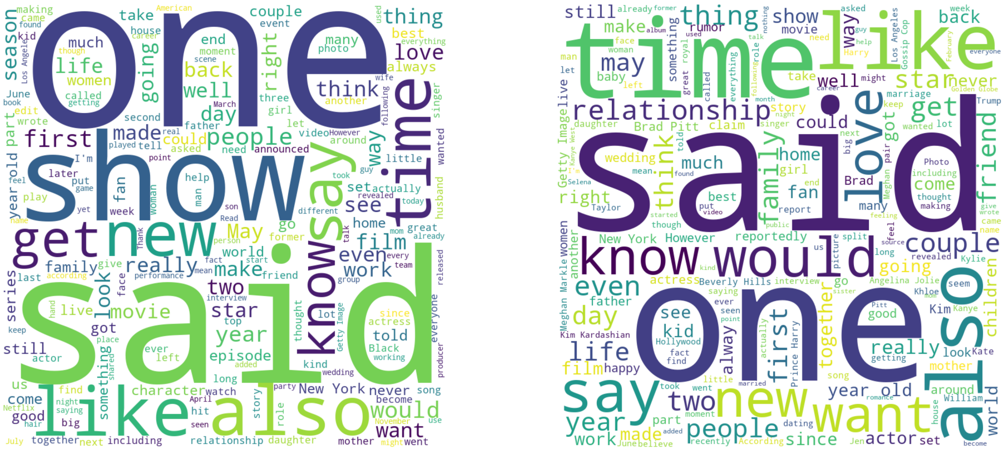
Figure 4.4: Typical examples of images associated with fake and real news in the PolitiFact dataset.

4.3.2 GossipCop

Figure 4.5 depicts the word clouds generated from the GossipCop dataset. The word cloud associated with fake news reveals prominent terms such as *said*, *one*, and *time*. Similarly, these words are also prevalent in the word cloud of real news, as shown in Figure 4.5a. However, a noteworthy distinction is the higher frequency of the word *show* in the real news subset. This observation suggests a larger proportion of news articles in the real category are related to television programming and shows.

Compared to the PolitiFact dataset, the distinction in word clouds between fake and real news articles is not as pronounced in the case of GossipCop. This finding implies that relying solely on textual content would be less effective in differentiating between fake and real news within the domain of entertainment news.

GossipCop



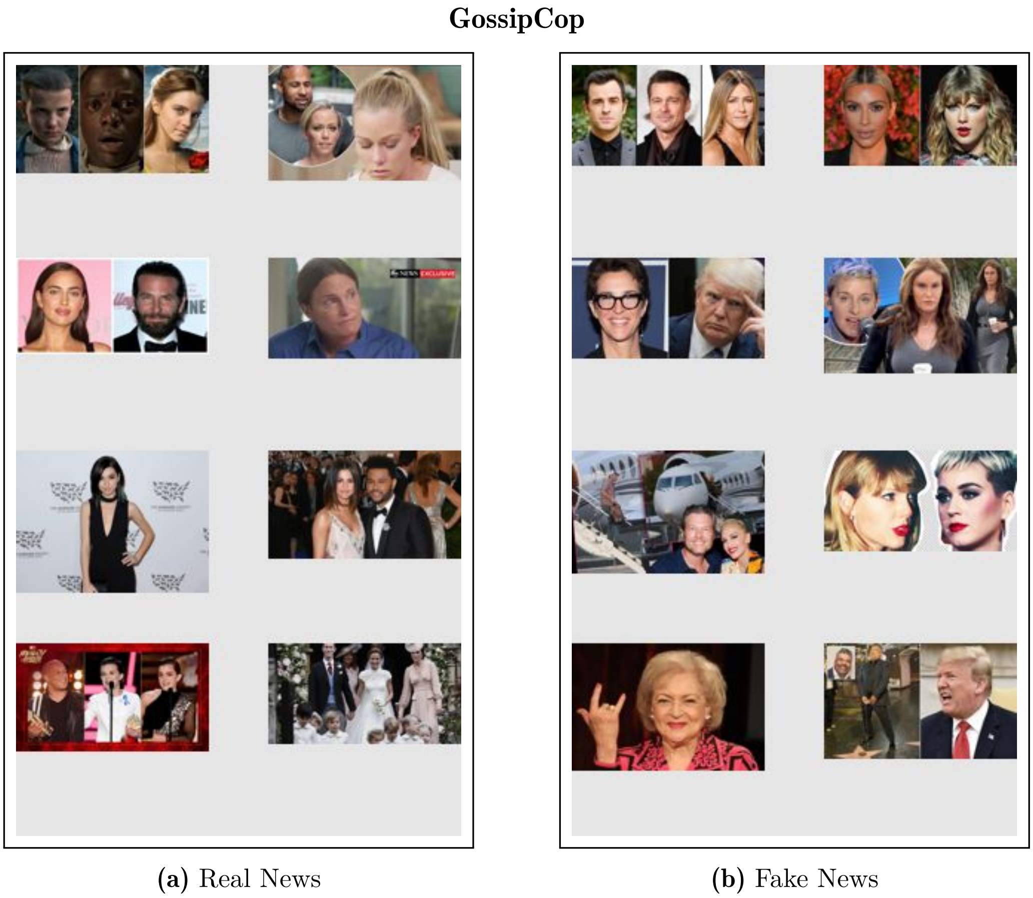
(a) Real News

(b) Fake News

Figure 4.5: Word clouds depicting the most frequent words in the GossipCop dataset, differentiating between real and fake news articles.

Shifting our attention to the analysis of images associated with fake and real news articles within the GossipCop dataset, we can observe typical examples in Figure 4.6. These images predominantly feature celebrities captured in diverse scenarios, including red-carpet events. Notably, within the subset of fake news, there is a relatively higher occurrence of images presenting celebrities in opposition to one another. These images often portray the faces of celebrities being juxtaposed, accompanied by dramatic facial expressions. For instance, a noteworthy example is an image where Katy Perry and Taylor Swift are depicted with seemingly intense expressions directed at each other. This suggests that fake news within the entertainment domain tends to emphasize and amplify dramatic elements in its visual presentation, in similarity to the fake news of PolitiFact.

Upon examining the images associated with fake and real news articles in the GossipCop dataset, although there is some variation, the distinction between the two types of news is not as pronounced as in the PolitiFact dataset. Both fake and real news articles utilize images of celebrities in a range of contexts, including images where celebrities are presented together. However, there is a noticeable difference in the nature of these images. In the real news subset, the celebrities' expressions tend to be more neutral and less confrontational compared to the exaggerated and intense expressions commonly seen in fake news images. While there are some visual cues that may hint at a distinction between real and fake news in the GossipCop dataset, the overall differentiation based solely on the images is not as clear-cut as in the PolitiFact dataset.



(a) Real News

(b) Fake News

Figure 4.6: Typical examples of images associated with fake and real news in the GossipCop dataset.

CHAPTER
FIVE

METHOD

Tackling the detection of fake news is a complex challenge that demands a comprehensive strategy. Previous approaches have explored various methods, each with their own strengths and weaknesses. This thesis presents an innovative architecture that combines the analysis of text and images from news articles to enhance the detection of fake news.

The chosen model to build upon is the KAHAN model, renowned for its efficient and innovative design. This model utilizes a hierarchical attention mechanism to process news articles and identify key sentences and words that indicate the presence of fake news. However, a limitation of the KAHAN model is its exclusive focus on text, disregarding the visual elements of news articles.

To overcome this limitation, we propose the Image-enhanced Knowledge-Aware Hierarchical Attention Network (I-KAHAN). This architecture extends the capabilities of the KAHAN model by incorporating image analysis, which is increasingly important in the realm of modern news. The primary objective of I-KAHAN is to improve fake news detection by integrating this crucial visual component.

The subsequent sections of this chapter will provide a detailed examination of I-KAHAN, including the alternative methods employed for image embedding, dimensionality reduction, and feature fusion within the architecture.

5.1 The I-KAHAN Architecture

The I-KAHAN architecture contemplates three attributes of news, as illustrated in Figure 5.1.

In this figure, the components inherited from KAHAN appear within a gray box, while the white area holds the new modules related to image integration. It also showcases the feature fusion mechanism responsible for merging these features and the classifier that determines the news classification.

The textual processing in I-KAHAN, represented within the gray area, follows a similar pattern as in the KAHAN model. It begins with embedding the article text (designated as *News Text*), using GloVe to convert the sentences S_i, comprising W_{ix}

words, into a numerical format. This phase also involves padding the sentences and words to a fixed length for a more structured representation. HAN then processes this embedded text to generate a concise and efficient representation. This operation utilizes the *Knowledge Extraction* component, which provides entities E_n and related entity claims C_{nm} from the knowledge base. Both of these processes were further elaborated in Chapter 2. The entities and claims are embedded using wiki2vec and contribute to the overall attention mechanism as illustrated by the yellow arrows in the figure. The outcome of these operations is a vector with 200 numerical values, which is then fed into the fusion component.

Similarly, the *Comments*, represented as a set of comments C_j containing sentences S_{jy}, undergo similar processing. For the sake of simplicity, the sub-event division process, which enables the model to consider the comments' timeline, is encapsulated under the *Comments Embed* operation. This operation also embeds the comments using GloVe and results in another 200-value vector.

The modules related to image integration occupy the left-hand side of Figure 5.1, within the white area. The raw image is initially processed by the *Transform* module to normalize, scale, and crop the pixel array, resulting in a more compact 3×224×224 matrix representation. This pre-processing is vital to ensure that the inputs to the subsequent embedding methods maintain consistent dimensions, regardless of the original image size.

The color-coded components in the figure represent image embedding (*Image Embed*), dimensionality reduction (*Dimensionality Reduction*), and feature fusion (*Feature Fusion*). These components employ multiple alternative methods, categorized under corresponding colors as detailed in the figure's top-left lists. For instance, *Image Embed* utilizes both CNN-based and CLIP-based methods, depicted in blue and purple, respectively.

Image Embed: This step transforms images into low-dimensional numerical representations or embeddings.

CNN-based: This category encompasses embeddings achieved through deep convolutional networks, specifically VGG19 and ResNet-50.

CLIP-based: This approach employs the CLIP to provide a common vector space representation of the text and image. A variant that includes entity attention on the CLIP embedding was also implemented.

Additional information on the image embedding methods can be found in Section 5.2.

Dimensionality Reduction: This operation condenses the large vectors generated by the CNN-based image embedding methods into a more manageable representation.

Pooling-based: This method uses max pooling and average pooling to compress the embeddings.

Neural net-based: This group includes a fully-connected layer and a deep neural network to reduce the size of the embedding. It offers a more sophisticated dimensionality reduction method due to the trainable parameters.

IHAN-based: This method includes a standalone and attention-enabled version of the proposed Image-based Hierarchical Attention Network (IHAN), an architecture inspired by the HAN approach.

More details on dimensionality reduction methods are available in Section 5.3.

Feature Fusion: This final component of the I-KAHAN architecture merges the feature vectors into a single representation compatible with the classifier.

Concatenation: This method combines features by stacking them.

Elementwise multiplication: This method multiplies the values of the feature vectors for fusion.

Averaging: This approach averages the values of the feature vectors for fusion.

Additional information on the methods for feature combination can be found in Section 5.4.

The bottom part of Figure 5.1 displays two different length image vectors. The first, depicted in blue, contains 200 numerical values, while the second carries 512 values. These colors correlate to their associated image embedding groups. Importantly, the CLIP-based embeddings do not require subsequent dimensionality reduction since they already produce relatively small vectors.

After feature fusion, a 200-length vector is forwarded to the classifier, a feed-forward neural network similar to that in the KAHAN model. This classifier performs binary classification, outputting a probability distribution over the two classes *fake* or *real.* Two alternative classifier architectures have been implemented, one with only one hidden layer, as in the KAHAN model, and another with two.

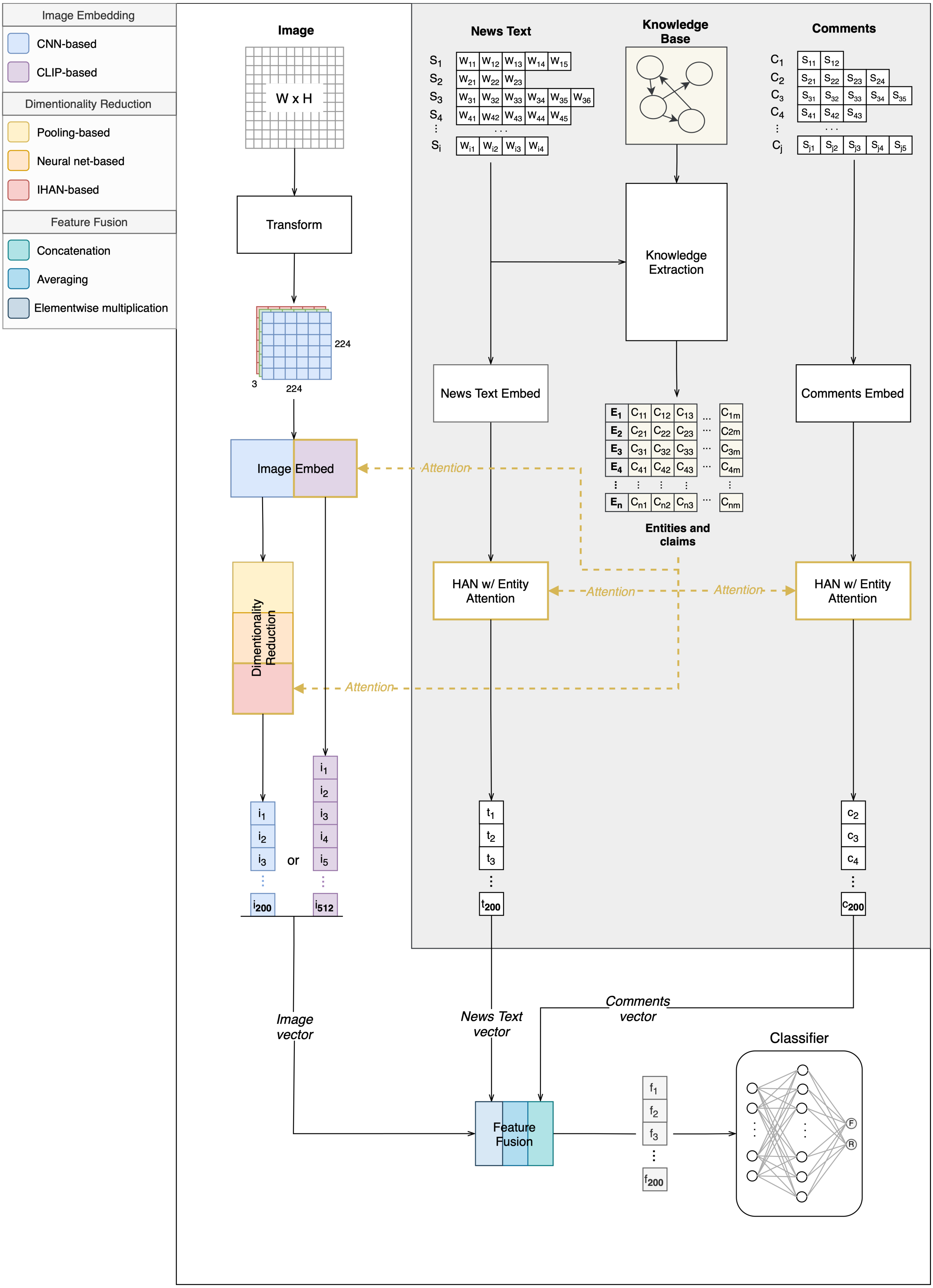


Figure 5.1: Overview of the I-KAHAN architecture

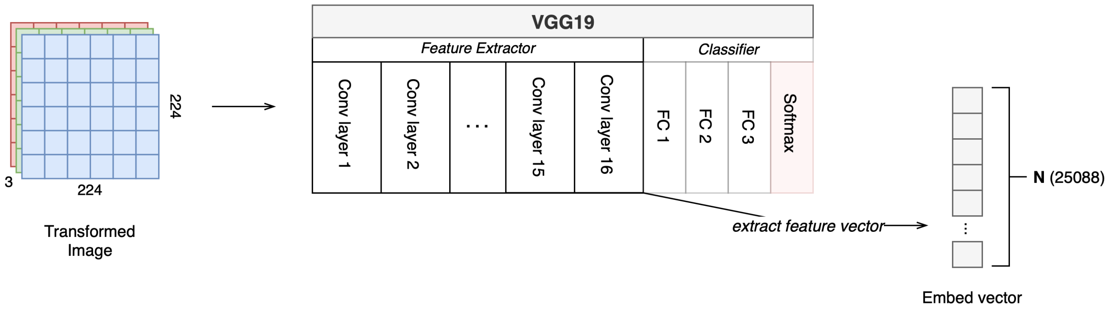
5.2 Image Embedding Techniques

This thesis leverages various image embedding techniques to generate numerical representations of images. Two primary categories of image embeddings are employed: pretrained deep CNNs, namely VGG19 and ResNet-50, and utilization of the CLIP model.

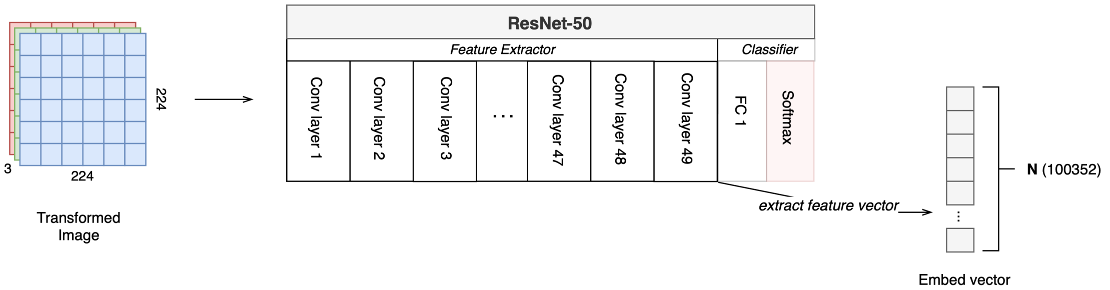
5.2.1 Deep CNN-based Image Embeddings

VGG19 and ResNet-50, popular deep CNN architectures, are adopted in this study for their proven efficacy in object recognition tasks, as well as their frequent use in the field of fake news detection, as discussed in Chapter 3.

As illustrated in Figure 5.2, both VGG19 and ResNet-50 perform feature extraction by generating numerical representations of images. The feature vector, denoted as N, is obtained from the final convolutional layer of each model, bypassing the usual softmax-activated vector of probabilities used in object classification tasks.



(a) VGG19-based feature extraction. A transformed image is passed to VGG19 for embedding, from which a feature vector (N) of size 25088 is collected by skipping the classification stage.



(b) ResNet-50-based feature extraction. Similar to VGG19, ResNet-50 embeds the transformed image, extracting a feature vector (N) of size 100352 by bypassing the classification phase.

Figure 5.2: Deep CNN-based image embedding strategies.

5.2.2 CLIP-based Image Embeddings

CLIP represents an innovative approach to image embedding with a relatively unexplored potential in the fake news detection domain. With its unique ability to encode images and text within a shared vector space, CLIP produces concise and information-rich embeddings that parallel textual features.

Figure 5.3 demonstrates two implementations of CLIP within this study's architecture. In the first instance (see Figure 5.3a), the transformed image is directly encoded through CLIP, yielding an image vector situated within a vector space. In the second approach, CLIP's text encoder is employed to separately embed entities and claims, following which multi-head attention is applied to the image vector. This innovative adaptation, inspired by the Hierarchical Attention Network (HAN), enables the image vector to align more closely with the most relevant claims, thereby boosting the representational power of the image.

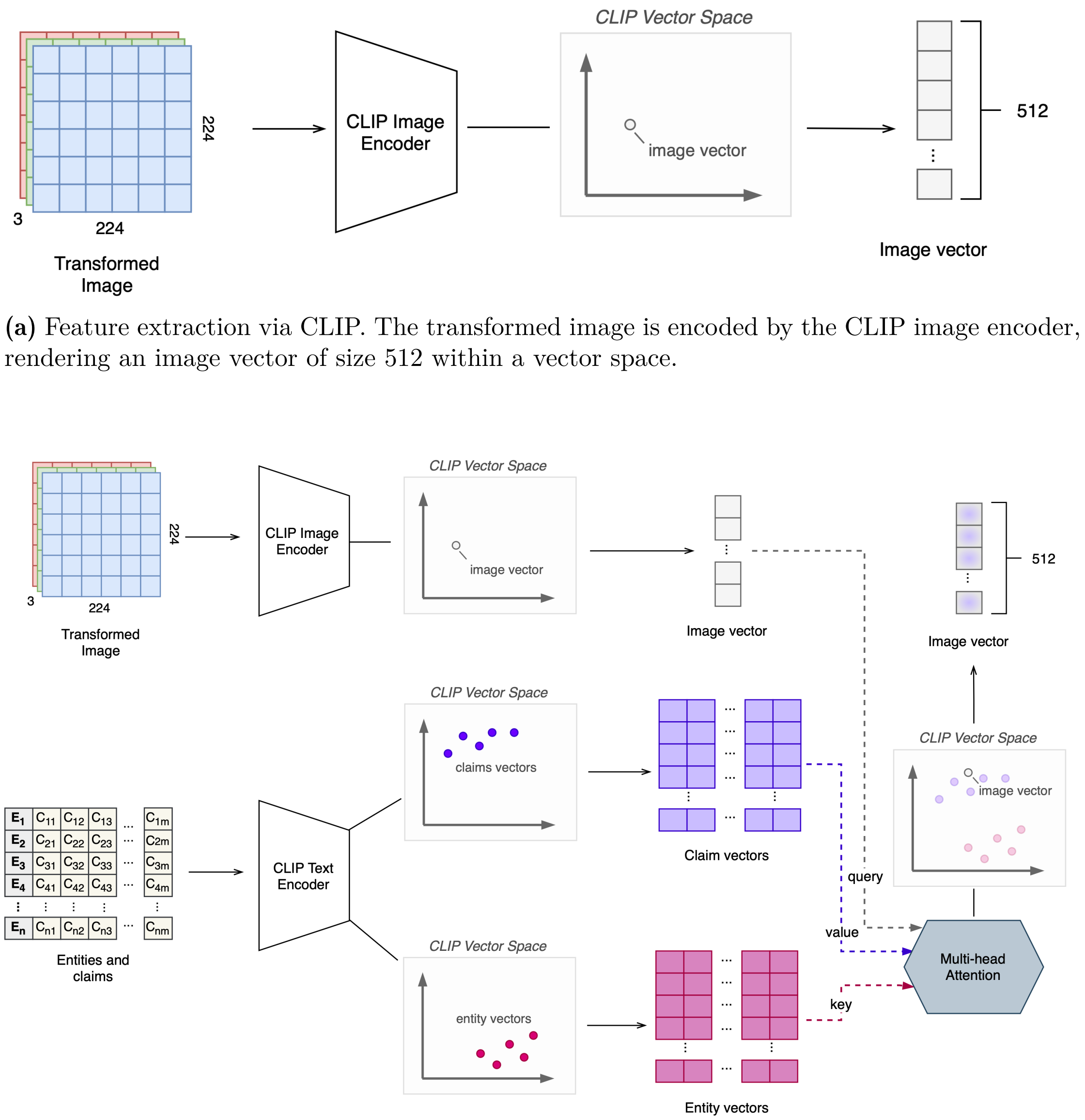
In summary, the chosen image embedding techniques comprise the well-established VGG19 and ResNet-50 deep CNN models, complemented by the novel CLIP-based embeddings. The selection was based on the successful use of these methods in the field of fake news detection, their capacity to generate meaningful image representations, and the innovative potential of CLIP, particularly when combined with entity attention inspired by the HAN model.

5.3 Dimensionality Reduction Methods

The large vectors produced by CNN-based embeddings necessitate a dimensionality reduction process, in order to reduce the size. Without this reduction, the dominance of high-dimensional image vectors (25,088 or 100,352 values) against the comparatively small textual vectors (200 values) could potentially skew the classifier's decisions. To deal with the imbalance, this thesis explores three categories of dimensionality reduction mechanisms: neural network-based methods, pooling-based techniques, and the novel Image-based Hierarchical Attention Network (IHAN) approach. The choice of these methods aligns with the goal of developing a robust and adaptive solution, capable of generating compressed yet informative image representations.

5.3.1 Neural Network-based Dimensionality Reduction

Neural networks, commonly employed for vector size reduction in related research, exhibit high representational power and can learn to produce increasingly accurate representations. As depicted in Figure 5.4, the image vectors derived from CNN-based image embeddings serve as the input for these networks, outputting a reduced image vector of size 200. Both networks, shown in Figures 5.4a and 5.4b, utilize the same number of input and output neurons. The difference lies in the network architecture: the latter

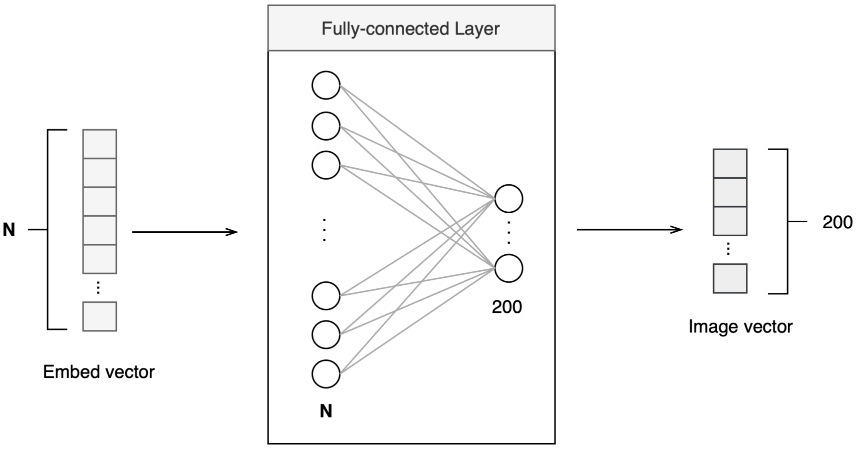


(a) Feature extraction via CLIP. The transformed image is encoded by the CLIP image encoder, rendering an image vector of size 512 within a vector space.

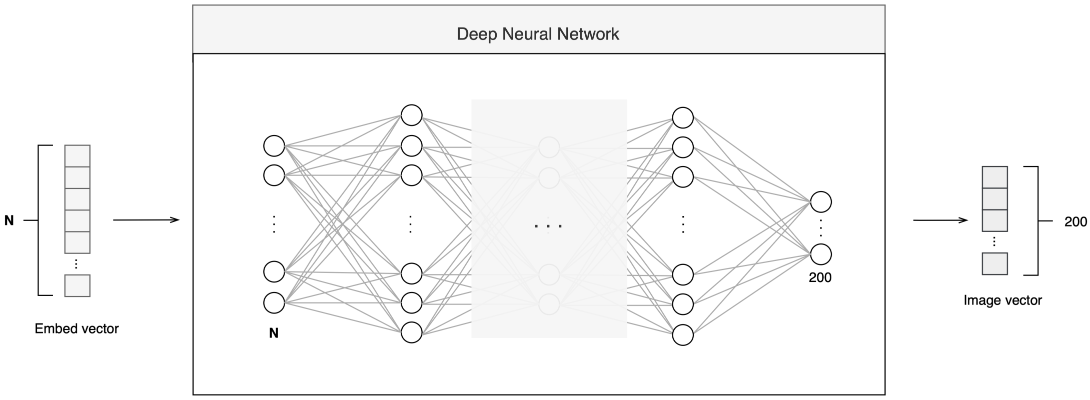
(b) CLIP-enhanced image encoding with multi-head attention. While the image is encoded as in the previous approach, entities and claims are separately embedded using CLIP's text encoder. This results in vectors representing the average claim for each entity, alongside entity vectors, all situated within the same vector space as the image. These vectors act as key (entity vectors) and value (claim vectors) in the multi-head attention mechanism, with the image vector as the query. Consequently, the representation of the image is adjusted to align more closely with the claims, enhancing its semantic richness while maintaining its original size.

Figure 5.3: Implementations of CLIP for image encoding.

incorporates multiple hidden layers, providing a richer representational power through its trainable parameters, but adding complexity to the training process.



(a) Fully-connected layer for dimensionality reduction. CNN-derived embeddings serve as the input, with a reduced image vector as the output.

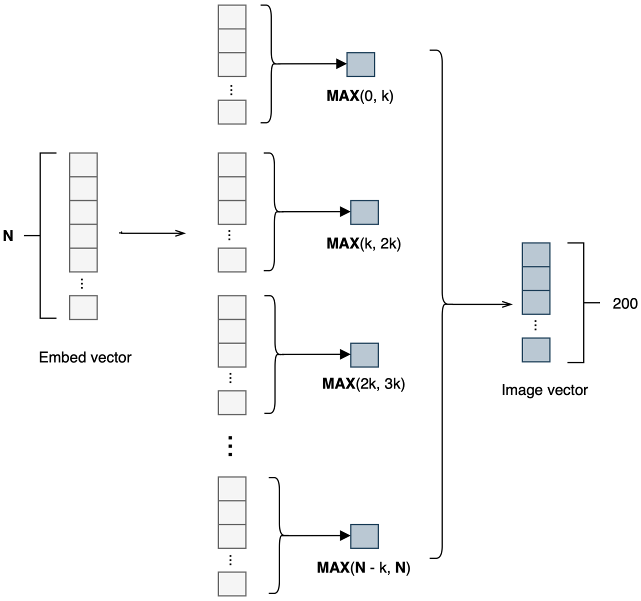


(b) Deep neural network for dimensionality reduction. The network consists of multiple hidden layers, allowing for greater representational power.

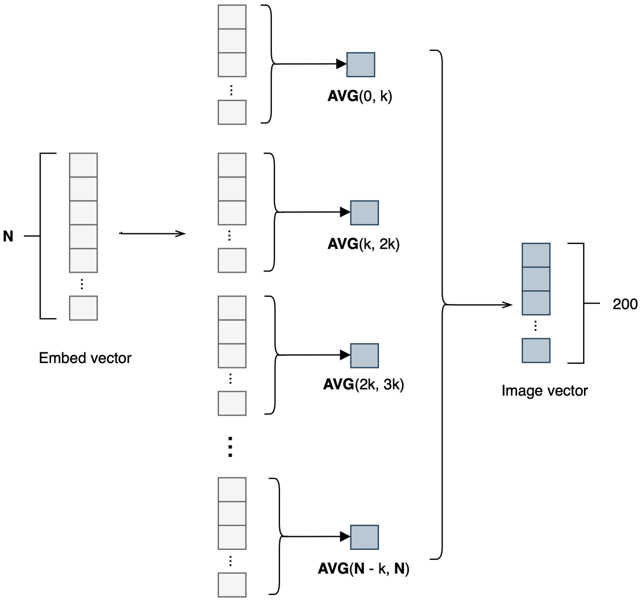
Figure 5.4: Neural network-based approaches for dimensionality reduction.

5.3.2 Pooling-based Dimensionality Reduction

As a simpler alternative, pooling techniques such as max pooling and average pooling were employed. These techniques divide the original vector into equal-sized patches, taking the maximum or average value from each patch to yield the reduced vector. This process, analogous to pooling in CNNs but applied to one-dimensional vectors in our case, is showcased in Figure 5.5. Although simpler and faster, these techniques may not deliver the representational power of neural networks as they are non-trainable.



(a) Max pooling process applied to the image embeddings. The embedding is partitioned into N segments of length k, with the maximum value from each segment forming the reduced image vector.



(b) Average pooling process applied to the image embeddings. The embedding is partitioned into N segments of length k, with the average value from each segment forming the reduced image vector.

Figure 5.5: Max and average pooling methods for dimensionality reduction.

5.3.3 Image-based Hierarchical Attention Network

Building upon the successful utilization of the HAN for textual content representation, this thesis introduces the Image-based Hierarchical Attention Network (IHAN) for visual content. The core premise of IHAN, similar to HAN, is the hierarchical structure of data: analogous to how words form sentences and sentences form text, smaller segments of an image contribute to the understanding of larger segments and, ultimately, the entire image. As such, IHAN is considered an intelligent, multi-level pooling method that extends the principles of HAN to image processing.

Furthermore, an additional attention layer, inspired by the entity-attention approach of the KAHAN model, was incorporated into IHAN to refine image embeddings by focusing on entities and claims similar to those found in the text content. This attention layer is expected to assist the classifier by providing more contextually-relevant visual information.

Figure 5.6 presents the implementation of IHAN in the architecture, with 5.6a showing the basic IHAN and 5.6b illustrating the enhanced version with the entity attention layer.

5.4 Feature Fusion Techniques

As one of the key components in the I-KAHAN architecture (see Figure 5.1), feature fusion serves to combine multiple feature vectors into a single representation. Although KAHAN used concatenation for this purpose, the inclusion of an additional modality in this work prompted the exploration of different techniques: concatenation, element-wise multiplication, and averaging. Each of these methods offers a unique approach to blending the features from text, comments, and images, denoted as x_n, x_c, and x_i, respectively.

Concatenation

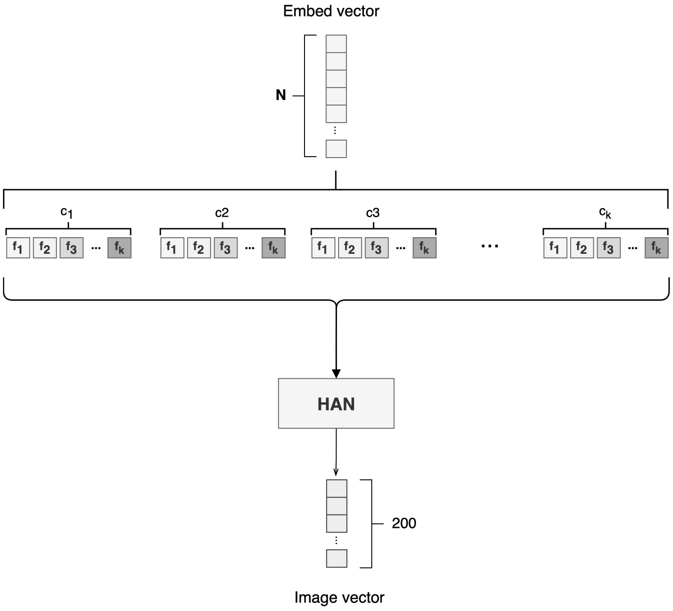
Concatenation simply stacks the feature vectors to form a new vector x_{concat}:

$$x_{concat} = x_n \oplus x_c \oplus x_i = \begin{bmatrix} x_n \\ x_c \\ x_i \end{bmatrix} \tag{5.1}$$

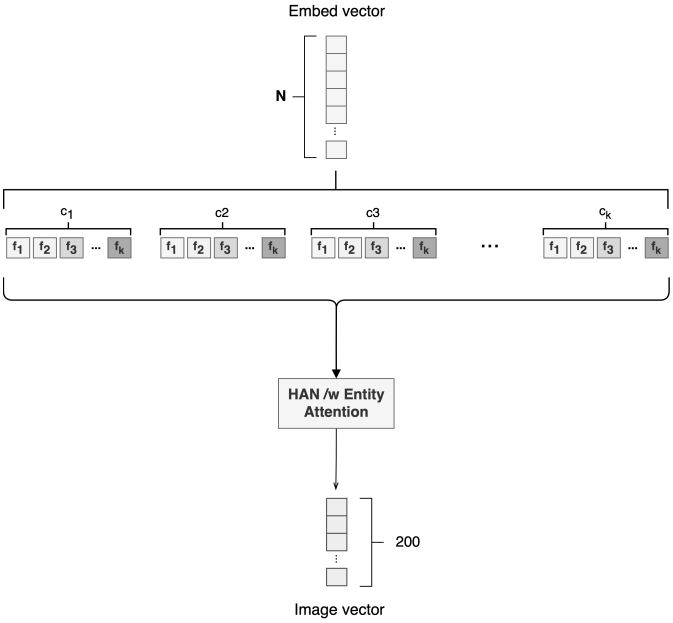
This technique maintains all the original information, albeit at the cost of increasing the dimensionality of the final representation. To solve this issue a fully-connected layer with 200 output neurons was applied to the output.

Element-wise Multiplication

Element-wise multiplication merges the feature vectors by multiplying their corresponding elements, forming a new vector x_{elem}:



(a) IHAN as an extension of HAN. The image embeddings are processed similarly to text in HAN, forming a hierarchical structure.



(b) IHAN with entity attention. The attention mechanism is designed to refine image embeddings by focusing on entities and claims found in the text.

Figure 5.6: The IHAN dimensionality reduction method, both in its standard form and with the addition of entity attention.

$$x_{elem} = x_n \odot x_c \odot x_i \tag{5.2}$$

This operation has the advantage of reinforcing the areas where all modalities agree, potentially emphasizing important features. However, it also risks obscuring information where only one or two modalities provide significant input.

Averaging

Averaging computes the element-wise mean of the feature vectors to form a new vector x_{avg}:

$$x_{avg} = \frac{1}{3} \sum (x_n, x_c, x_i) \tag{5.3}$$

This method effectively compromises between the other two techniques: it reduces dimensionality like element-wise multiplication but maintains the notion of distinct input sources like concatenation. However, it may dilute the impact of individual features.

CHAPTER
SIX

EXPERIMENTS

This chapter presents the experiments carried out to evaluate and compare different image integration techniques in the I-KAHAN architecture. Three principal tasks, namely embedding generation, dimensionality reduction, and feature fusion, are the core focus. The experiments utilize the GossipCop and PolitiFact datasets and assess a variety of I-KAHAN configurations. The chosen evaluation metrics include accuracy, precision, recall, F1 score, as well as the confusion matrix.

The chapter further highlights the key tools and technologies employed, such as PyTorch, Torchvision, IDUN, Open CLIP, Gensim, Scikit-learn, and NLTK, and provides a clear outline of the experimental setup. Details about the chosen hyperparameters and how the experiments were executed are also discussed.

6.1 Experimental Design

The set of competing techniques presented in the previous chapter was thoroughly examined in order to determine the best-performing methods for each of the three tasks associated with image integration, namely embedding generation, dimensionality reduction, and feature fusion, and in addition a comparative experiment between the shallow and deep architecture of the classifier. To achieve a solid understanding of the strengths and shortcomings of each technique, various configurations of I-KAHAN, taking into account every possible combination of techniques, was trained and evaluated on the GossipCop and PolitiFact datasets. Furthermore, an additional experiment was conducted to evaluate the data quality of the modified dataset collection process, namely FakeNewsNet+, with its original counterpart, FakeNewsNet.

Outlined below is a comprehensive list of the experiments that were carried out. The scheme consists of three principal experiments, the first of which is labeled as Experiment 1, further broken down into four sub-experiments marked as Experiment 1A, Experiment 1B, Experiment 1C, and Experiment 1D. The remaining experiment is designated as Experiment 2.

Experiment 1 This experiment encapsulates the four sub-experiments **(A)**, **(B)**, **(C)** and **(D)**.

- **(A)** This experiment focuses on the difference in classification performance between the various embedding generation techniques.
- **(B)** In this experiment, the performance of the individual dimensionality reduction methods is evaluated.
- **(C)** This experiment explores the performance of the different feature fusion methods.
- **(D)** This examines the performance of the enhanced classifier as compared to the original.

Experiment 2 This final experiment assesses the quality of data collected through the improved process (FakeNewsNet+) against the original (FakeNewsNet).

Four metrics have been employed as the basis for evaluating these experiments, namely accuracy, precision, recall, and the F1 score. In addition, the confusion matrix has been employed for a comprehensive evaluation of the overall classification performance of the proposed architecture. The details of these metrics and the confusion matrix were discussed in Chapter 2.

6.2 Tools and Technologies

The following subsections detail the specific tools and technologies employed in conducting the experiments of this study. These comprise various programming languages, software libraries, and hardware resources. Each tool was selected on the basis of its robust capabilities, wide acceptance in the scientific community, and its relevance to the tasks at hand.

PyTorch[1] PyTorch is an open-source machine learning library from the Facebook AI Research team. It provides a flexible deep learning framework and encourages efficient research prototyping and development. PyTorch supports tensor computation with strong GPU acceleration and automatic differentiation for building and training neural networks. In this thesis, it was employed to perform all machine learning-related tasks, including the implementation of a learning rate scheduler that dynamically adjusts the learning rate during training to help reduce overfitting and improve model generalization.

Torchvision[2] Torchvision, a PyTorch add-on, provides access to popular datasets, model architectures, and image transformations for computer vision.

[1] `https://pytorch.org`

[2] `https://pytorch.org/vision/stable/index.html`

It simplifies the process of loading data and includes functionalities for tasks such as reading images and applying transformations. In this research, it was employed to perform image transformations and to load the VGG19 and ResNet-50 models.

IDUN[3] IDUN is a High-Performance Computing (HPC) service offered by the Norwegian University of Science and Technology. With its powerful computational resources, it's a pivotal tool for large-scale data processing and complex computations. In this research, the IDUN system was harnessed for executing exhaustive experiments and training comprehensive models. The array jobs function of IDUN was particularly beneficial for managing parallel processes. Furthermore, the SLURM workload manager was utilized for submitting, monitoring, and managing jobs.

Open CLIP[4] Open CLIP, an open-source rendition of OpenAI's CLIP model, is a neural network that leverages a vast variety of internet text. For this research, Open CLIP was employed for encoding entities, claims, and images, thus enriching the representational capacity of the developed models.

Gensim[5] Gensim is an open-source Python library designed to handle large text collections with data streaming and incremental algorithms. For the purpose of this research, it was used to download GloVe pretrained embeddings and to load these embeddings into the model, hence facilitating efficient and effective textual data processing.

Scikit-learn[6] Scikit-learn is a versatile machine learning library for Python that offers a wide array of algorithms for classification, regression, clustering, and dimensionality reduction. It also provides tools for model fitting, data preprocessing, model selection, and evaluation, which were utilized extensively throughout the various stages of this research.

NLTK[7] The Natural Language Toolkit (NLTK) is a leading platform for constructing Python programs to work with human language data. With its comprehensive suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, NLTK was employed for the rigorous preprocessing of textual data in this study.

[3] https://www.hpc.ntnu.no/idun/

[4] https://github.com/mlfoundations/open_clip

[5] https://radimrehurek.com/gensim/index.html

[6] https://scikit-learn.org/stable/

[7] https://www.nltk.org

6.3 Experimental Setup

This section describes the experimental setup, providing the necessary information and resources to replicate the experiments conducted in this study, with further details found in the associated repository. The KAHAN[8] codebase served as the starting point for these experiments, and it underwent considerable modifications and enhancements to incorporate image features. In addition, the code was optimized for performance to facilitate comprehensive experimentation within the time frame of the thesis. The revised I-KAHAN codebase[9] is available on GitHub[10].

6.3.1 Prerequisite Data

Certain pre-trained models, namely GloVe and Wikipedia2vec, are required by the I-KAHAN architecture for text embedding purposes. Table 6.1 outlines these prerequisite data files. The GloVe models, *glove-wiki-gigaword-100* for news content and *glove-twitter-100* for comments, were sourced via the Gensim library. The *enwiki-20180420-100d.pkl* file, used for embedding the entities and entity claims through Wikipedia2vec, was obtained from the Hugging Face[11] platform.

Unlike text embeddings, image embeddings did not require any external downloads. The Torchvision library's Models subpackage provided access to the VGG19 and Resnet-50 model, while the Open CLIP library was used to get access to the *ViT-B-32-quickgelu* pre-trained model used to implement the CLIP model.

Table 6.1: Pre-trained data files required for textual embeddings, including their sources and uses.

Pre-trained File	**Usage**	**Source**
glove-wiki-gigaword-100	Embedding news content	Gensim's GloVe model
glove-twitter-100	Embedding comments	Gensim's GloVe model
enwiki-20180420-100d.pkl	Embedding entities and entity claims	Hugging Face

6.3.2 Hyperparameters

The hyperparameters for the I-KAHAN model are presented in Table 6.2. They have been kept consistent across all experiments to ensure an equitable comparison among

[8] `https://github.com/ienlie0513/KAHAN`

[9] `https://github.com/oysteinlondal/I-KAHAN`

[10] `https://github.com`

[11] `https://huggingface.co`

different configurations. The selection of these parameters was an iterative process based on initial experiments and refined to maximize the model's classification performance.

The hyperparameters from the original KAHAN model served as the starting point for these initial experiments. For instance, parameters such as batch size and weight decay were found to be optimal at their original values from the KAHAN experiments and thus were carried forward into the I-KAHAN model.

6.3.3 Execution of the Experiments

Each experiment was run as a separate array job on the IDUN system, with each experiment comprising 39 distinct jobs per dataset. This number originates from the application of the two embedding models (VGG19 and ResNet-50), each tested with the six dimensionality reduction techniques and the three fusion methods discussed in Chapter 5. Two additional jobs account for the application of the CLIP model and the CLIP model enhanced with entity attention, which employs only the concatenation fusion technique and requires no dimensionality reduction. Finally, the KAHAN model execution serves as a baseline comparison.

In total, the experiments amount to $39 \times 4 \times 2 = 312$ unique jobs, where 39 signifies the configurations of I-KAHAN per dataset, 4 accounts for the two datasets (GossipCop and PolitiFact) collected using two different platforms (FakeNewsNet and FakeNewsNet+), and 2 signifies the use of two distinct classifiers, namely the shallow and the deep.

Table 6.2: Hyperparameters utilized in all I-KAHAN model experiments.

Hyperparameter	Value	Explanation
Epochs	65	Number of times the model iteratively learns from the data.
Batch Size	16	Number of samples the model learns from in one iteration.
Learning Rate	5×10^{-5}	Rate at which model parameters are updated during training.
Number of Seeds	3	Number of repetitions for cross-validation to enhance reliability.
Number of Folds	3	Number of partitions of the data in k-fold cross-validation.
Hidden Size	100	Size of the first hidden layer of the classifier influencing model complexity.
Weight Decay	1×10^{-4}	Coefficient for L2 regularization, preventing overfitting.
Dropout	0.3	Probability of a neuron being temporarily ignored during training, aiding in preventing overfitting.

CHAPTER

SEVEN

RESULTS

The outcomes of the experiments outlined in the prior chapter will be presented and elucidated in this chapter. Detailed results of each individual experiment will be discussed initially, followed by a comparison of the top-performing I-KAHAN configurations on the two datasets, PolitiFact and GossipCop, against the baseline. The data presented are based on the average scores across all *k*-fold cross-validation folds and repetitions for each I-KAHAN configuration. This methodology ensures the maximum accuracy and reliability of the results, thereby mitigating any possible impact from chance or randomness.

7.1 Experiment 1: Comparison of Alternative Methods

This section presents the outcomes of the first experiment, with each sub-experiment's results elaborated through separate plots and tables. The first three sub-experiments utilize a process of averaging the scores of configurations using the same methods, providing a fair basis for comparing the performance of individual methods.

7.1.1 (A) Image Embedding Methods

Figure 7.1 depicts the average F1 score achieved by each embedding technique across all configurations. The blue and orange bars represent the PolitiFact and GossipCop datasets, respectively. The plot reveals that the methods tend to perform better on the PolitiFact dataset, although this is not of primary concern for this experiment due to other factors such as data quality discrepancies. However, it is noteworthy that the rankings are consistent across both datasets. For example, CLIP scores the highest F1 in both datasets, followed by CLIP with entity attention, while ResNet-50 and VGG19 perform similarly on both datasets.

Table 7.1 provides further insight into the performance of the methods. It compares performance across the metrics used during the experimentation. The best-performing methods are highlighted in bold, with the second-best underlined. Across both datasets,

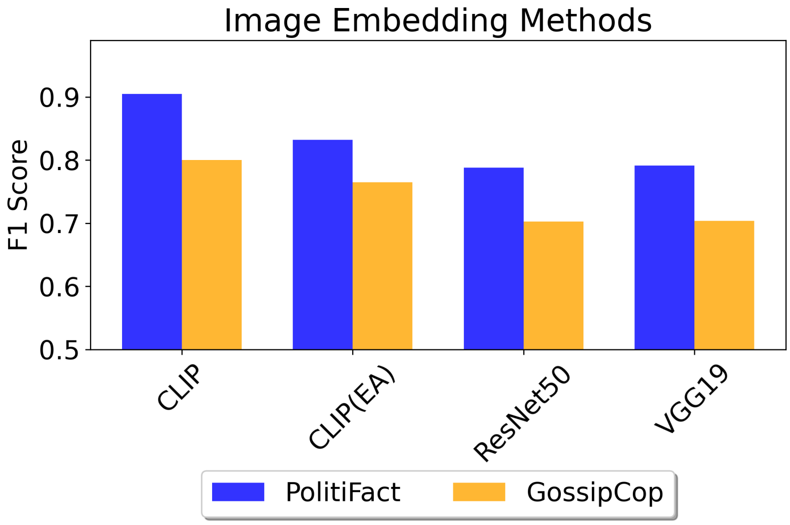


Figure 7.1: Comparison of the image embedding methods on the PolitiFact and GossipCop datasets. The F1 score has been employed as the metric for the comparison. The blue marks the F1 score on PolitiFact, while the yellow marks those of GossipCop.

CLIP outperforms the other methods, with the difference being most significant on the PolitiFact dataset.

7.1.2 (B) Dimensionality Reduction Methods

The findings of Experiment 1B, which examines different dimensionality reduction techniques, are presented in Figure 7.2 and Table 7.2. As per the blue and orange bars in the figure, the techniques are consistently more effective on the PolitiFact dataset, and the ranking order appears to be relatively similar across both datasets. One clear inference is the relatively weak performance of the fully-connected layer, but the top performer is less evident from the chart alone.

Table 7.2 offers a more detailed perspective, revealing that max pooling outperforms other methods on the PolitiFact dataset, closely trailed by average pooling. Interestingly, the deep neural network approach does marginally better on the precision metric for this dataset. On the GossipCop dataset, however, IHAN followed by average pooling offer the best scores. Across both datasets, the three methods IHAN, max pooling, and average pooling demonstrate highly similar performance, indicating their interchangeable effectiveness.

Table 7.1: Comparison of image embedding methods on PolitiFact and GossipCop. The F1 score is employed as the metric for the comparison. The blue and orange bars represent the F1 scores on PolitiFact and GossipCop datasets, respectively.

(a) PolitiFact

Method	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP	**0.9059**	**0.9090**	**0.9025**	**0.9049**
CLIP(EA)	0.8373	0.8391	0.8345	0.8323
ResNet50	0.7945	0.7951	0.7952	0.7881
VGG19	0.7984	0.8000	0.7974	0.7914

(b) GossipCop

Method	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP	**0.8011**	**0.8020**	**0.8001**	**0.8005**
CLIP(EA)	0.7657	0.7669	0.7650	0.7649
ResNet50	0.7115	0.7219	0.7139	0.7026
VGG19	0.7129	0.7236	0.7151	0.7037

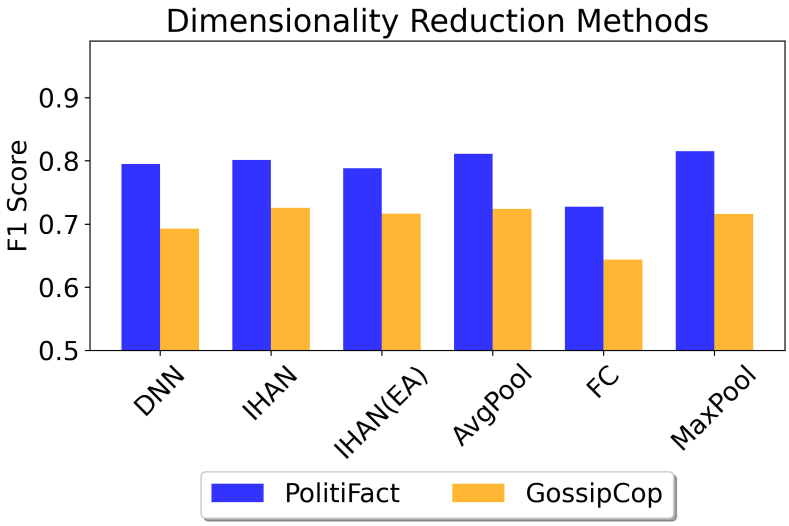


Figure 7.2: Comparison of dimensionality reduction methods on the PolitiFact and GossipCop datasets. The F1 score is employed as the metric for the comparison. The blue and orange bars represent the F1 scores on PolitiFact and GossipCop datasets, respectively.

Table 7.2: Comparison of dimensionality reduction methods on PolitiFact and GossipCop datasets. The highest numbers are in bold, while those underlined are the second highest. Across both datasets, max pooling, IHAN, and average pooling perform comparably well.

(a) PolitiFact

Method	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
DNN	0.8015	0.8196	0.7955	0.7947
IHAN	0.8124	0.7899	0.8091	0.8015
IHAN(EA)	0.7971	0.7824	0.8022	0.7883
AvgPool	0.8168	0.8118	0.8159	0.8114
FC	0.7325	0.7503	0.7384	0.7275
MaxPool	**0.8183**	**0.8312**	**0.8166**	**0.8151**

(b) GossipCop

Method	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
DNN	0.7116	0.6745	0.7064	0.6927
IHAN	**0.7310**	**0.7548**	**0.7371**	**0.7258**
IHAN(EA)	0.7240	0.7474	0.7280	0.7167
AvgPool	0.7296	0.7540	0.7353	0.7242
FC	0.6549	0.6604	0.6535	0.6435
MaxPool	0.7220	0.7455	0.7267	0.7160

7.1.3 (C) Feature Fusion Techniques

Figure 7.3 demonstrates the average F1 score of each fusion technique across all configurations. In line with previous results, the fusion methods perform better on the PolitiFact dataset. The fusion methods consistently rank in the same order on both datasets. Particularly, concatenation proves to be the superior fusion method, followed by averaging, while element-wise multiplication lags behind significantly.

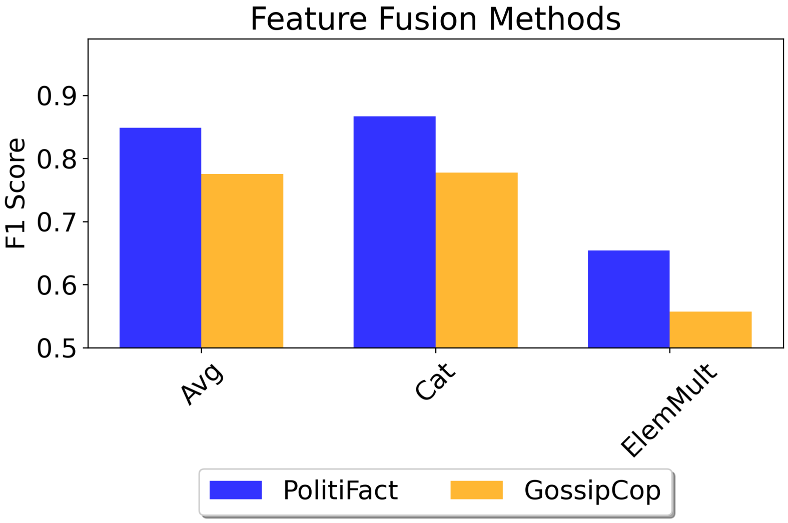


Figure 7.3: Evaluation of feature fusion methods on the PolitiFact and GossipCop datasets using F1 scores. Blue bars represent the PolitiFact dataset, while orange bars indicate the GossipCop dataset.

Table 7.3 delivers a more comprehensive perspective on the performance metrics of the fusion techniques. It emphasizes the negligible performance difference between concatenation and averaging, especially on the GossipCop dataset. However, it's worth noting that the element-wise multiplication method performs significantly poorer than its counterparts. In fact, on the GossipCop dataset, configurations employing element-wise multiplication for feature fusion yield results only slightly better than a random guessing strategy.

7.1.4 (D) Classifier Comparison

This section is dedicated to the comparative analysis of two classifiers' performance: a shallow classifier with a single hidden layer, and a more complex model with an additional hidden layer. The goal is to identify the classifier that performs most effectively in accurately classifying the datasets used.

Table 7.3: Detailed comparison of feature fusion methods on the PolitiFact and GossipCop datasets. The highest values are indicated in bold, while the second-highest values are underlined. Concatenation achieves the best performance, closely followed by averaging.

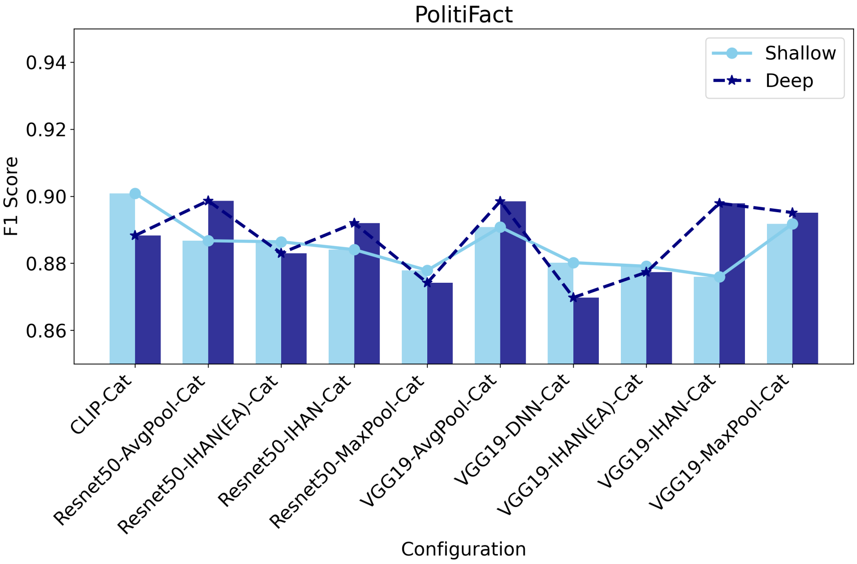
(a) PolitiFact

Method	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
Avg	<u>0.8500</u>	<u>0.8510</u>	<u>0.8467</u>	<u>0.8486</u>
Cat	**0.8683**	**0.8694**	**0.8665**	**0.8670**
ElemMult	0.6716	0.6731	0.6760	0.6539

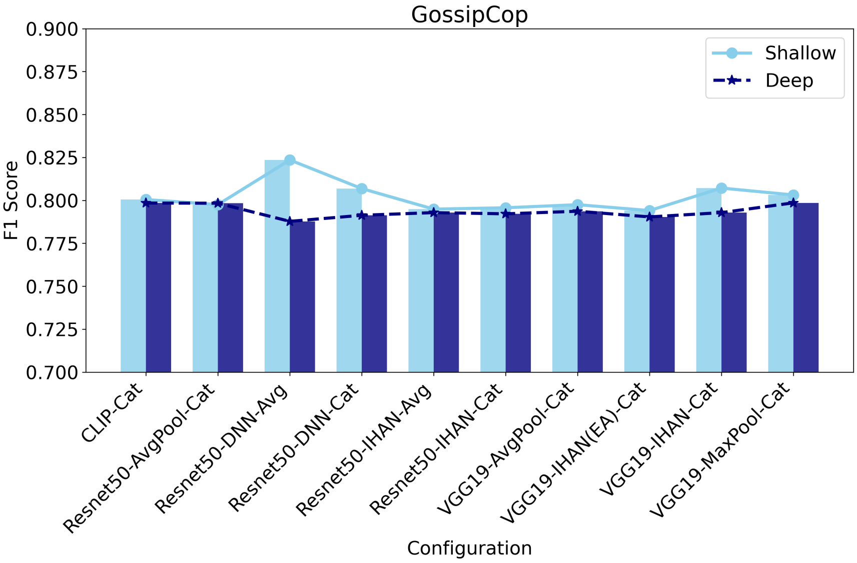
(b) GossipCop

Method	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
Avg	<u>0.7757</u>	<u>0.7763</u>	<u>0.7755</u>	<u>0.7752</u>
Cat	**0.7781**	**0.7788**	**0.7782**	**0.7776**
ElemMult	0.5836	0.6141	0.5906	0.5575

Figure 7.4 illustrates the classifiers' performance on the PolitiFact and GossipCop datasets. Light blue bars on the left-hand side of Figure 7.4a represent the performance of the shallow classifier, while the dark blue bars on the right-hand side depict the performance of the deeper model. A line graph is added to the bar plots to highlight the performance differences across all configurations. For conciseness, the graph includes only ten configurations, labeled on the x-axis. The label for each configuration is a combination of the embedding model, dimensionality reduction technique, and fusion method used, separated by dashes. For instance, a configuration using the CLIP model for image embedding, no reduction technique, and concatenation for feature combination is denoted as *CLIP-Cat*. Likewise, a configuration using ResNet-50 with IHAN for reduction and concatenation for fusion is denoted as *Resnet50-IHAN-Cat*. Figure 7.4b represents a similar plot for the GossipCop dataset.



(a) PolitiFact



(b) GossipCop

Figure 7.4: Performance comparison of the shallow and deep classifier across various configurations on the PolitiFact and GossipCop datasets.

The ten configurations were selected based on their superior average performance on each dataset when using the shallow classifier. The aim was to see if the deeper classifier could enhance the performance of these configurations. However, the results contradict this initial assumption. Especially in the GossipCop dataset, the shallow classifier generally matches or outperforms the deeper model. With the PolitiFact datasets, the best classifier varies across configurations, making it difficult to determine a clear winner. For instance, in the *VGG19-IHAN-Cat* configuration, the deeper classifier considerably improves performance, while for the *CLIP-Cat* configuration, it significantly reduces performance.

To provide a more detailed comparison, Table 7.4a and Table 7.4b below the plots represent the data from the graphs with enhanced precision. The numbers on the left represent the shallow classifier, and the numbers on the right represent the deep classifier, with the best results highlighted in bold. Table 7.4a provides a detailed comparison of the PolitiFact dataset. As indicated by the bold text, the shallow classifier and the deep classifier each excel an equal number of times, meaning neither classifier consistently outperforms the other. In contrast, Table 7.4b, detailing the results for the GossipCop dataset, demonstrates that the shallow classifier outperforms the deeper model in seven out of the ten configurations, suggesting that the shallow model is generally better suited for this dataset.

To review the full comparison, including all configurations, refer to the extended results presented in Appendix A.

7.2 Experiment 2: Comparing FakeNewsNet and FakeNewsNet+

The following section of the study presents the results of the comparison between the original FakeNewsNet data collection process and the enhanced version, referred to as FakeNewsNet+. The comparison is depicted through graphs and tabulated data for an in-depth analysis. The figures used in this evaluation represent the average of averages across all folds and seeds, for both datasets.

Figure 7.5 reveals an intriguing trend in the performance of different configurations across both datasets, PolitiFact and GossipCop. While there is minimal performance variance between FakeNewsNet and FakeNewsNet+ for most configurations, some show significant improvements with FakeNewsNet+. For instance, the configurations *VGG19-AvgPool-Cat* and *VGG19-DNN-Cat* applied to PolitiFact demonstrate an approximate 5% increase with the improved data collector. On the GossipCop dataset, the *Resnet50-DNN-Avg* and *Resnet50-DNN-Cat* configurations also show notable improvements with FakeNewsNet+, albeit less dramatic than those seen with PolitiFact. The *CLIP-Cat* configuration appears to perform slightly better with the original data collector, FakeNewsNet, but the difference is negligible at around 1%.

From the observed trends, it's evident that configurations utilizing deep neural networks, pooling operations, or IHAN for dimensionality reduction are likely to benefit from the data collector enhancements. Conversely, configurations that employ CLIP embeddings seem to experience a slight performance decrease, while others remain largely unaffected.

Table 7.4: Detailed comparison of the shallow and deep classifier on the PolitiFact and GossipCop datasets. The numbers on the left side of each slash are those of the shallow classifier, while those on the right-hand side are of the deep one. The highest numbers are presented in bold.

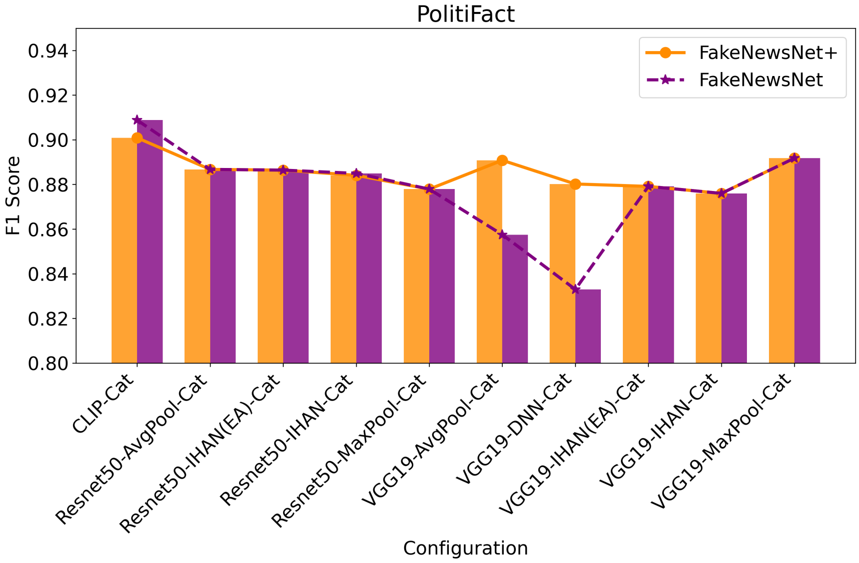
(a) PolitiFact

Configuration	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP-Cat	**0.9020**/0.8892	**0.9059**/0.8911	**0.8983**/0.8874	**0.901**/0.8883
Resnet50-AvgPool-Cat	0.8875/**0.8995**	0.8873/**0.8988**	0.8863/**0.8980**	0.887/**0.8987**
Resnet50-IHAN(EA)-Cat	**0.8876**/0.8841	**0.8911**/0.8842	**0.8849**/0.8809	**0.886**/0.8830
Resnet50-IHAN-Cat	0.8850/**0.8926**	0.8859/**0.8922**	0.8836/**0.8923**	0.884/**0.8920**
Resnet50-MaxPool-Cat	**0.8790**/0.8756	**0.8796**/0.8777	**0.8770**/0.8715	**0.878**/0.8742
VGG19-AvgPool-Cat	0.8918/**0.8994**	0.8922/**0.8997**	0.8888/**0.8966**	0.891/**0.8985**
VGG19-DNN-Cat	**0.8816**/0.8721	**0.8854**/0.8809	**0.8772**/0.8631	**0.880**/0.8698
VGG19-IHAN(EA)-Cat	**0.8799**/0.8781	**0.8798**/0.8771	**0.8793**/0.8777	**0.879**/0.8773
VGG19-IHAN-Cat	0.8773/**0.8986**	0.8789/**0.8976**	0.8734/**0.8983**	0.876/**0.8979**
VGG19-MaxPool-Cat	0.8926/**0.8960**	0.8924/**0.8968**	0.8911/**0.8940**	0.892/**0.8951**

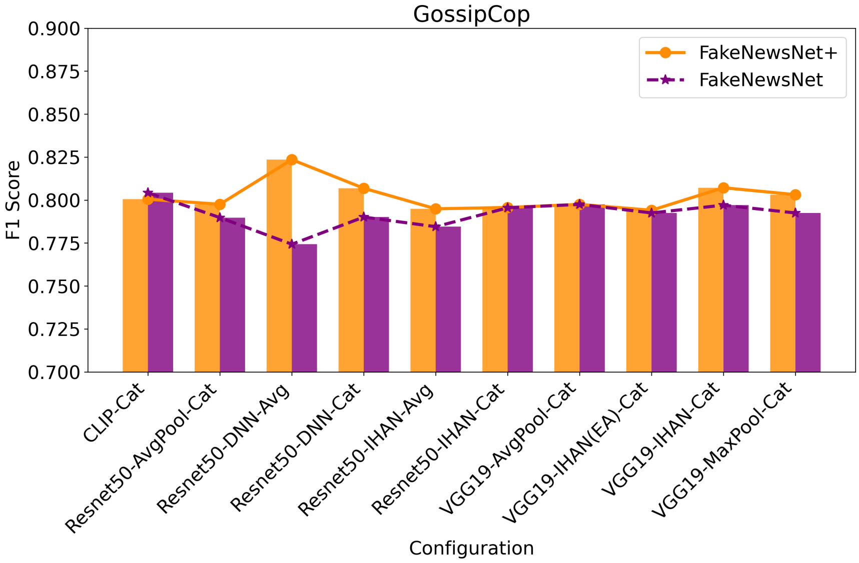
(b) GossipCop

Configuration	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP-Cat	**0.8011**/0.7987	**0.8020**/0.7986	**0.8001**/0.7991	**0.800**/0.7984
Resnet50-AvgPool-Cat	0.7978/**0.7984**	0.7973/**0.7999**	0.7980/**0.8005**	0.798/**0.7983**
Resnet50-DNN-Avg	**0.8242**/0.7881	**0.8247**/0.7919	**0.8222**/0.7898	**0.824**/0.7878
Resnet50-DNN-Cat	**0.8073**/0.7920	**0.8075**/0.7950	**0.8069**/0.7916	**0.807**/0.7913
Resnet50-IHAN-Avg	**0.7950**/0.7930	**0.7955**/0.7940	**0.7964**/0.7945	**0.795**/0.7929
Resnet50-IHAN-Cat	**0.7959**/0.7924	**0.7965**/0.7937	**0.7968**/0.7935	**0.796**/0.7921
VGG19-AvgPool-Cat	**0.7977**/0.7938	**0.7978**/0.7938	**0.7984**/0.7948	**0.797**/0.7937
VGG19-IHAN(EA)-Cat	**0.7944**/0.7904	**0.7963**/0.7926	**0.7954**/0.7930	**0.794**/0.7904
VGG19-IHAN-Cat	**0.8073**/0.7930	**0.8084**/0.7934	**0.8090**/0.7942	**0.807**/0.7928
VGG19-MaxPool-Cat	**0.8034**/0.7987	**0.8041**/0.7995	**0.8044**/0.8002	**0.803**/0.7986

Figure 7.5: Comparative performance of configurations using FakeNewsNet and FakeNewsNet+



(a) PolitiFact



(b) GossipCop

Performance results are also presented in a tabular format for detailed comparison in Table 7.5. It offers a comparative analysis of FakeNewsNet+ and FakeNewsNet for the PolitiFact (Table 7.5a) and GossipCop (Table 7.5b) datasets. For each metric, the left and right numbers denote the performance scores of FakeNewsNet+ and FakeNewsNet respectively. An extensive dataset of the results can be found in the appendix referenced in the previous section.

Table 7.5 reveals that, for the PolitiFact dataset, there are equal instances where a configuration performs best under both FakeNewsNet+ and FakeNewsNet. However, it's critical to note that when FakeNewsNet outperforms, it does so by a small margin. In contrast, when FakeNewsNet+ leads, the performance difference is significant. Following the trends observed in Experiment 1D, FakeNewsNet+ outperforms FakeNewsNet for most configurations. For the few configurations where FakeNewsNet+ is not as effective, the margin is relatively small. These findings underscore the benefits of the enhanced FakeNewsNet+ data collector, as it consistently delivers comparable or better results across a wide range of configurations and metrics.

The same trend can be observed in the GossipCop dataset, albeit to a lesser extent. FakeNewsNet+ and FakeNewsNet seem to compete neck-to-neck, each outperforming the other in equal instances. Again, the margin of outperformance by FakeNewsNet+ is typically larger, particularly when it comes to F1 scores.

7.3 Overall Performance

The comparative performance of various configurations on the PolitiFact and GossipCop datasets is apparent from the findings outlined earlier. As shown in Figure 7.6, the PolitiFact dataset consistently yields higher performance than the GossipCop dataset, with the difference exceeding 5% for most configurations. The blue bars on the left represent the performance of each configuration on the PolitiFact dataset, while the orange bars on the right denote the performance on the GossipCop dataset.

The line graphs suggest a trend of inverse performance between the two datasets. For instance, the configuration *Resnet-50-DNN-Avg* is optimal for GossipCop but is the least efficient for PolitiFact. Likewise, *CLIP-Cat* yields good results for PolitiFact but falters slightly on GossipCop.

Table 7.5: Performance comparison between FakeNewsNet+ and FakeNewsNet for different configurations. The numbers on the left side of each slash represent FakeNewsNet+, while those on the right-hand side represent the original FakeNewsNet. The highest numbers are in bold.

(a) PolitiFact

Configuration	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP-Cat	0.9020/**0.9097**	0.9059/**0.9121**	0.8983/**0.9067**	0.901/**0.9088**
Resnet50-AvgPool-Cat	0.8875/0.8875	0.8873/0.8873	0.8863/0.8863	**0.887**/0.8867
Resnet50-IHAN(EA)-Cat	0.8876/0.8876	0.8911/0.8911	0.8849/0.8849	0.886/**0.8864**
Resnet50-IHAN-Cat	0.8850/**0.8858**	0.8859/**0.8866**	0.8836/**0.8845**	0.884/**0.8850**
Resnet50-MaxPool-Cat	0.8790/0.8790	0.8796/0.8796	0.8770/0.8770	**0.878**/0.8779
VGG19-AvgPool-Cat	**0.8918**/0.8595	**0.8922**/0.8597	**0.8888**/0.8543	**0.891**/0.8575
VGG19-DNN-Cat	**0.8816**/0.8364	**0.8854**/0.8426	**0.8772**/0.8272	**0.880**/0.8330
VGG19-IHAN(EA)-Cat	0.8799/0.8799	0.8798/0.8798	0.8793/0.8793	0.879/**0.8791**
VGG19-IHAN-Cat	0.8773/0.8773	0.8789/0.8789	0.8734/0.8734	0.876/0.8760
VGG19-MaxPool-Cat	0.8926/0.8926	0.8924/0.8924	0.8911/0.8911	**0.892**/0.8918

(b) GossipCop

Configuration	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP-Cat	0.8011/**0.8046**	0.8020/**0.8045**	0.8001/**0.8044**	0.800/**0.8042**
Resnet50-AvgPool-Cat	**0.7978**/0.7903	**0.7973**/0.7903	**0.7980**/0.7896	**0.798**/0.7898
Resnet50-DNN-Avg	**0.8242**/0.7751	**0.8247**/0.7759	**0.8222**/0.7736	**0.824**/0.7743
Resnet50-DNN-Cat	**0.8073**/0.7907	**0.8075**/0.7907	**0.8069**/0.7894	**0.807**/0.7901
Resnet50-IHAN-Avg	**0.7950**/0.7847	**0.7955**/0.7856	**0.7964**/0.7859	**0.795**/0.7845
Resnet50-IHAN-Cat	**0.7959**/0.7957	**0.7965**/0.7963	**0.7968**/0.7966	**0.796**/0.7955
VGG19-AvgPool-Cat	0.7977/0.7977	0.7978/0.7978	0.7984/0.7984	0.797/**0.7975**
VGG19-IHAN(EA)-Cat	**0.7944**/0.7927	**0.7963**/0.7941	**0.7954**/0.7940	**0.794**/0.7925
VGG19-IHAN-Cat	**0.8073**/0.7972	**0.8084**/0.7981	**0.8090**/0.7985	**0.807**/0.7970
VGG19-MaxPool-Cat	**0.8034**/0.7928	**0.8041**/0.7935	**0.8044**/0.7932	**0.803**/0.7925

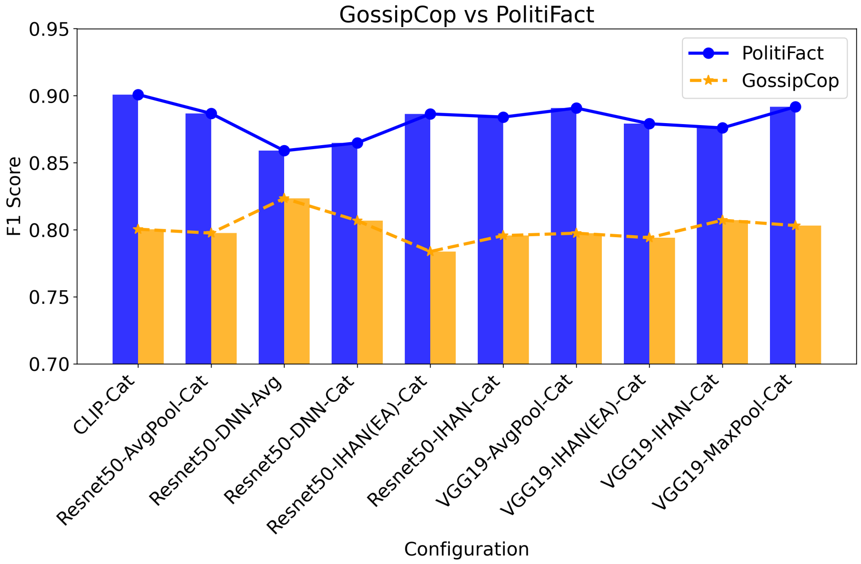


Figure 7.6: Performance comparison of configurations on the PolitiFact and GossipCop datasets.

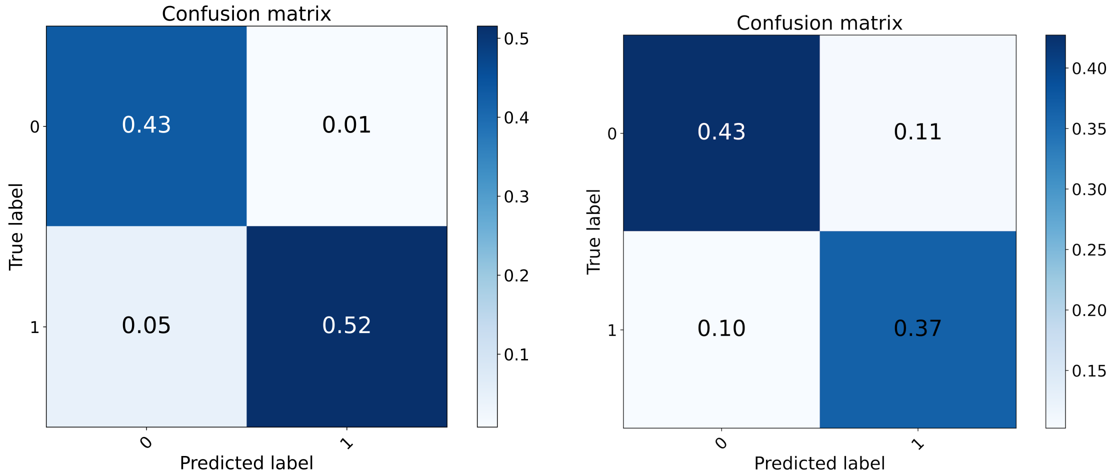
Table 7.6 presents a more detailed overview of the performance of each configuration. The best results for each dataset are emphasized in bold, with blue representing PolitiFact and orange for GossipCop. The second-best results are underlined. *CLIP-Cat* and *Resnet50-DNN-Avg* are the best configurations for PolitiFact and GossipCop, respectively, and are highlighted in the leftmost column of the table. The configuration that achieved the highest average score across both datasets, *CLIP-Cat*, is highlighted with a border.

Table 7.6: Comparison of the overall performance of various configurations across both datasets.

Configuration	**PolitiFact vs GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP-Cat	**0.9020**/0.8011	**0.9059**/0.8020	**0.8983**/0.8001	**0.901**/0.8005
Resnet50-AvgPool-Cat	0.8875/0.7978	0.8873/0.7973	0.8863/0.7980	0.887/0.7975
Resnet50-DNN-Avg	0.8611/**0.8242**	0.8674/**0.8247**	0.8541/**0.8222**	0.859/**0.8235**
Resnet50-DNN-Cat	0.8653/0.8073	0.8677/0.8075	0.8684/0.8069	0.865/0.8068
Resnet50-IHAN(EA)-Cat	0.8876/0.7840	0.8911/0.7853	0.8849/0.7852	0.886/0.7838
Resnet50-IHAN-Cat	0.8850/0.7959	0.8859/0.7965	0.8836/0.7968	0.884/0.7957
VGG19-AvgPool-Cat	0.8918/0.7977	0.8922/0.7978	0.8888/0.7984	0.891/0.7975
VGG19-IHAN(EA)-Cat	0.8799/0.7944	0.8798/0.7963	0.8793/0.7954	0.879/0.7941
VGG19-IHAN-Cat	0.8773/0.8073	0.8789/0.8084	0.8734/0.8090	0.876/0.8071
VGG19-MaxPool-Cat	0.8926/0.8034	0.8924/0.8041	0.8911/0.8044	0.892/0.8031

For a more in-depth understanding of the top-performing configurations, confusion matrices have been constructed for each, as depicted in Figure 7.7. Each matrix captures the classification performance of a given fold during k-fold cross-validation. Figure 7.7a represents the *CLIP-Cat* confusion matrix for PolitiFact, indicating a low number of false negatives and positives. Interestingly, true positives exceed true negatives by 52% to 43%. This trend is reversed in Figure 7.7b, where the *Resnet50-DNN-Avg* confusion matrix for GossipCop shows more true negatives than positives and slightly higher rates of false negatives and positives. This aligns with the data in Table 7.6.

In essence, these confusion matrices suggest that *CLIP-Cat* leans towards classifying news more often as real, while *Resnet50-DNN-Avg* tends to classify news more often as fake.



(a) Confusion matrix for *CLIP-Cat* on the PolitiFact dataset.

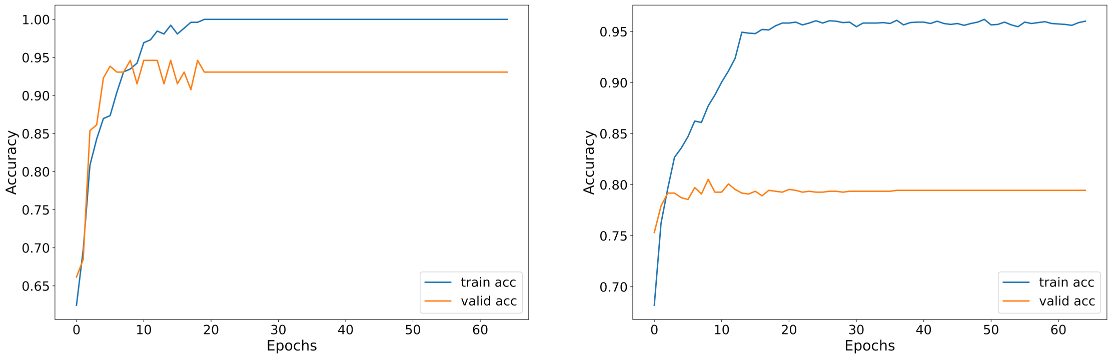
(b) Confusion matrix for Resnet50-DNN-Avg on the GossipCop dataset.

Figure 7.7: Confusion matrices for the top-performing configurations on the PolitiFact and GossipCop datasets.

7.3.1 Performance Details of the Best-Performing I-KAHAN Configuration

The superior configuration, as indicated in Table 7.6 and highlighted with dark borders, is *CLIP-Cat*. Given its overall superior performance, this configuration is expected to function well on real-world data. Therefore, it has been selected for the final I-KAHAN model, and a comprehensive analysis of its performance is provided below. This starts with an examination of the accuracy per epoch during training, as illustrated in Figure 7.8.

Figure 7.8a displays the accuracy of the training and validation sets of an arbitrary fold during the k-fold cross-validation of the configuration on the PolitiFact dataset. The graph shows a sharp increase in accuracy for the first 10 epochs, after which the increase slows for the remaining 55 epochs with no further changes in accuracy. The accuracy of the training set is nearly optimal, suggesting the model is fitting the training data too well. This can often indicate difficulties in classifying new data, although in this case, it is not a major concern due to the measures implemented to reduce the impact of this, particularly the learning rate scheduler. Also, this phenomenon is likely due to the small size of the PolitiFact dataset. The accuracy of the *CLIP-Cat* on the GossipCop dataset, as shown in Figure 7.8b, also plateaus after the initial few epochs, albeit at a lower level. The model, however, does not seem to overfit as much, with the training accuracy remaining around 95%. This may be due to the larger size of the GossipCop dataset. Overfitting seems to be mitigated by the learning rate scheduler for both datasets, although the large difference between training and validation accuracy is particularly noteworthy in the latter case.



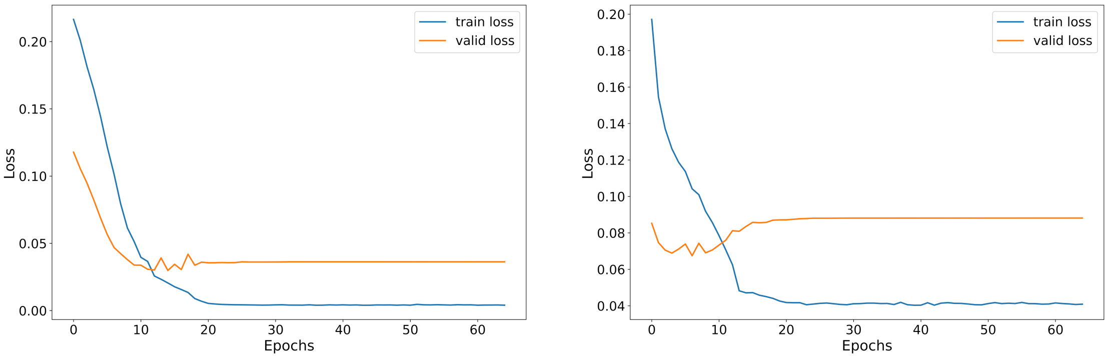
(a) Accuracy per epoch for *CLIP-Cat* on the PolitiFact dataset.

(b) Accuracy per epoch for *CLIP-Cat* on the GossipCop dataset.

Figure 7.8: Accuracy per epoch during training for the *CLIP-Cat* configuration.

Examining the loss per epoch graph provides another useful perspective on the performance of the configuration. Figure 7.9 displays the loss per epoch for *CLIP-Cat* on each dataset. In both cases, the validation loss starts lower than the training loss, which might seem counterintuitive as the model is trained on the training set and should therefore have a lower loss. However, this is most likely due to the implementation of both regularization and dropout on the classifier, which inflates the training loss. As shown in Figure 7.9a, the training loss of the model is nearing zero on the PolitiFact dataset. This indicates overfitting, although the learning rate scheduler seems to mitigate this as shown by the graphs leveling off. A similar observation can be made with the loss on the GossipCop dataset, as shown in Figure 7.9b. In this case, the model does not appear to overfit as much, maintaining a training loss slightly higher than zero. However, the validation loss does rise slightly after the initial few epochs and remains at that level.

The confusion matrices for *CLIP-Cat* on each dataset are presented in Figure 7.10. The left-most matrix, shown in Figure 7.10a, is identical to the one presented earlier since the same configuration performs best overall and on the PolitiFact dataset. The configuration generally performs well on both datasets. However, as illustrated in Figure 7.10b, the configuration is more conservative on the GossipCop dataset. This pattern

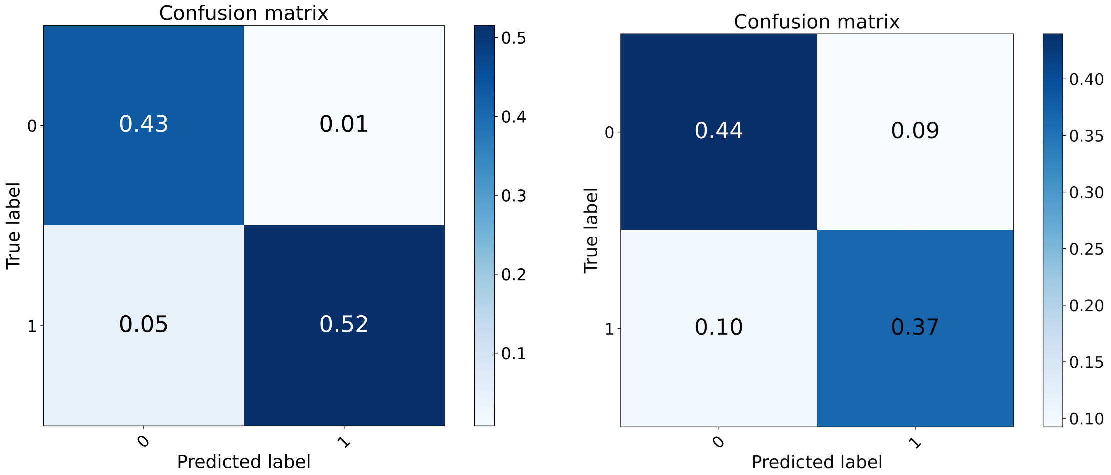


(a) Loss per epoch for *CLIP-Cat* on the PolitiFact dataset.

(b) Loss per epoch for *CLIP-Cat* on the GossipCop dataset.

Figure 7.9: Loss per epoch during training for the *CLIP-Cat* configuration.

was also observed for the *Resnet50-DNN-Avg* configuration, suggesting this is due to the dataset's imbalance, with a higher proportion of fake news. Conversely, the PolitiFact dataset has a higher proportion of real news, explaining why the configuration is comparatively more optimistic.



(a) Confusion matrix for *CLIP-Cat* on the PolitiFact dataset.

(b) Confusion matrix for *CLIP-Cat* on the GossipCop dataset.

Figure 7.10: Confusion matrices for the *CLIP-Cat* configuration on each dataset.

7.3.2 Comparison of I-KAHAN with Baseline KAHAN

Table 7.7 presents the performance of the KAHAN model for each of the four metrics, with the I-KAHAN model displayed below for comparison. As previously stated, the I-KAHAN model utilizes the *CLIP-Cat* configuration due to its superior performance.

Upon examination, it becomes apparent that the I-KAHAN model outperforms the baseline across all metrics. However, the difference is not significant, especially not for the GossipCop dataset. On the GossipCop dataset, the I-KAHAN model surpasses the KAHAN model by slightly over 1% for all metrics, while on the PolitiFact dataset, it shows a slightly more substantial improvement of approximately 3%.

Table 7.7: Comparison of overall performance between I-KAHAN and baseline KAHAN.

	PolitiFact				GossipCop			
	Accuracy	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
KAHAN	0.8756	0.8762	0.8732	0.8745	0.7894	0.7904	0.7905	0.7892
I-KAHAN	**0.9020**	**0.9059**	**0.8983**	**0.901**	**0.8011**	**0.8020**	**0.8001**	**0.8005**

CHAPTER

EIGHT

EVALUATION AND DISCUSSION

This chapter evaluates the performance of the I-KAHAN model for fake news detection. It delves into the nuances behind the results and discusses the research process, highlighting the challenges encountered, the methodological considerations, and the ethical implications of the work.

The previous chapter presented the experimental results of the two experiments conducted in this thesis, namely Experiment 1, with sub-experiments 1A-D, and Experiment 2. The evaluation of the first experiment is presented in the next section, while the other one is in the section after that.

8.1 Assessing Experiment 1: Evaluating the Alternative Methods and Classifier Variations

In this section, the results of Experiment 1 will be evaluated and discussed. The discussion has been divided into two, where the results of Experiment 1A-C are detailed first, followed by that of 1D.

8.1.1 Experiment 1A-C Evaluation: Impact of Different Methods on Performance

The results of the first three sub-experiments (1A-C) demonstrate intriguing findings about the various methods and their impacts on the I-KAHAN model's performance. While the experiments validate certain hypotheses about the effectiveness of some methods, they also expose unexpected behavior from others.

Assessment of Image Embedding Performance

As was seen through the results presented in the previous chapter, the CLIP-based image embeddings performed around 10% better than the CNN-based embeddings on both the PolitiFact and GossipCop dataset. One reason could be the benefit of CLIP's ability to represent both text and image data in a common vector space, which may aid the classifier in correlating the text and visuals more easily.

This added advantage of CLIP may be particularly useful in light of the issues surrounding fact-checking images inherent in the PolitiFact dataset, as discussed in Chapter **??**. Specifically, some of the real news in the dataset included images embedded with text affirming their authenticity, while certain fake news featured images annotated with text indicating that they had been fact-checked and deemed false. These factors could contribute to the improved performance of CLIP-based image embeddings.

Interestingly, CLIP-based embeddings perform over 10% better on PolitiFact than on GossipCop. One reason could be the prevalence of embedded text in PolitiFact's images, which CLIP, unlike CNN-based methods, can effectively process. However, when entity attention was introduced to the CLIP embeddings, the performance degraded significantly. One hypothesis for this could be due to the technical constraints which limited entity attention to just the top 10 entities and a maximum of 25 entity claims each, compared to 100 entities for the text embeddings.

The scope of the experiment could have been expanded to consider the same number of entities and claims as the text content, although this would require overcoming the performance and resource limitations currently faced when using the CLIP encoder on the IDUN cluster.

Comparative Analysis of Dimensionality Reduction Methods

For Experiment 1B, the best method of dimensionality reduction differed between the datasets. Max pooling was best on PolitiFact, and IHAN was best on GossipCop. In general, simpler methods like pooling operations outperformed more complex ones like the deep neural network approach. This might suggest that, for this context, simpler representations are more effective.

Interestingly, the use of entity attention in IHAN degraded the performance. This could be because while the text and entity-claim vectors share the same vector space, the CNN-based image embeddings and the entity-claim vectors do not. This may have limited the effectiveness of entity attention.

Evaluating Fusion Methods

In Experiment 1C, the element-wise multiplication method performed worst on both datasets, while averaging and concatenation were close to the top. This suggests that element-wise multiplication might not be the optimal fusion method in this context. It is possible that element-wise multiplication, by introducing non-linearities, may have resulted in a more complex representation that the classifier struggles to learn from.

Summary of Findings from Experiments 1A-C

The rankings from these experiments provide valuable insights, but they should be interpreted with caution. The reason being that the performance of each method was averaged across all configurations of I-KAHAN in which it was used, which might not fully reflect their potential under specific configurations. For instance, configurations with ResNet-50 and the deep neural network, despite their low individual rankings, were part of the best-performing configuration on GossipCop.

Moreover, the consistent rankings across both datasets suggest that the effectiveness of these methods might be largely independent of the dataset used, suggesting a broader implication of these results for future research in this field. However, the experiments should have been conducted on an even more comprehensive collection of diverse datasets to further confirm this.

8.1.2 Evaluating Experiment 1D: Comparative Analysis of Deep and Shallow Classifiers

In the comparative analysis of deep and shallow classifiers, the introduction of an additional hidden layer yielded interesting findings. The classifier's performance varied considerably depending on the dataset and configuration applied, illustrating the complex interaction between data attributes, model structure, and the learning process.

Improved performance was observed on the GossipCop dataset when the classifier incorporated an additional hidden layer, especially those employing IHAN or average pooling for dimensionality reduction, with the exception of instances incorporating entity attention with IHAN. This can be attributed to the enhanced capacity for learning more intricate patterns granted by the additional layer. However, an additional layer also increases the model's complexity, potentially leading to overfitting, particularly when paired with complex dimensionality reduction methods like the deep neural network approach.

Even with dropout and regularization techniques in place, the deep classifier did not exhibit the expected level of performance. A potential explanation for this behavior might be the slow-down in the learning process introduced by these features. Dropout and regularization aim to curtail overfitting, but in doing so, they extend the model's training time by adding a complexity penalty, in the case of regularization, or by randomly deactivating neurons during training, in the case of dropout. This additional time, compounded with the inherent complexity of some of the dimensionality reduction methods, might limit the efficiency of the classifier's learning process. Although the value of both of these hyperparameters was chosen based on some initial experimentation, it might have been beneficial to take it a step further by utilizing a more extensive search for optimal values, through the implementation of for example grid search. The grid search mechanism exhaustively searches through a predefined set of hyperparameter combinations to find the optimal configuration for a machine learning model [66].

8.2 Experiment 2 Analysis: Impact of Revised Dataset on Performance

This section will delve into the assessment of the dataset collected using the proposed FakeNewsNet+ collection process, and its impact on the performance of I-KAHAN. The evaluation aims to analyze the role and influence of data quality and its inherent influence on the performance of the different configurations.

8.2.1 Impact of Data Enhancement on Performance

The revised dataset played a crucial role in enhancing I-KAHAN's performance across different configurations. A diverse and high-quality dataset offers a rich learning context for the model, assisting in the formation of more accurate representations. However, the performance varied across different configurations, underlining the nuanced relationship between the datasets and the various model configurations. However, the general trend, as we observed in the previous chapter, is that the revised dataset at large led to a significant improvement in classification performance. Certain configurations exhibited a remarkable performance boost of 10% beyond that of the initial dataset.

8.2.2 Reflections on Dataset Limitations and Further Improvements

Dataset limitations and potential areas of improvement are critical aspects that need addressing. One of the primary challenges encountered during data collection was the inaccurate extraction of text content from news articles by the scraping tool. This resulted in the scraping of irrelevant data such as website details and cookie policies. Although these instances were eliminated during data cleaning, the issue highlights the need for a more sophisticated data scraping tool to improve the quality of collected data. For instance, the recent WorkGPT framework[1], showing impressing web scraping capabilities, could have been utilized.

Dataset size and balance pose significant challenges as well. The size of the PolitiFact dataset was noticeably reduced following the data cleaning, possibly affecting the results due to the smaller sample size. This underlines the need for alternative methods to augment the data volume, either through additional data sources or synthetic data augmentation techniques.

The imbalance of datasets raises concerns about the potential impact of the trade-off between data retention and balance. Techniques such as under-sampling, over-sampling, and Synthetic Minority Over-sampling Technique (SMOTE) could be considered to address this issue. Under-sampling reduces the size of the majority class while over-sampling increases the minority class size, and SMOTE synthesizes new examples in the minority class, thereby maintaining a balance without significantly reducing the dataset size [67].

Furthermore, the specificity of the PolitiFact and GossipCop datasets to political and celebrity news respectively limits the scope of the study. Incorporating a broader range

[1] `https://github.com/team-openpm/workgpt`

of news categories could offer more comprehensive insights and robust results.

8.3 Comparing I-KAHAN with the State-of-the-Art

Comparisons with state-of-the-art models are integral to research as they shed light on the relative performance of the developed model, which in this case is the I-KAHAN. This analysis helps to answer one of the research questions regarding how I-KAHAN compares with the latest advancements in multi-modal fake news detection.

Table 8.1 outlines the F1 scores of I-KAHAN in contrast to several state-of-the-art multi-modal fake news detection models. On the PolitiFact dataset, the FND-CLIP model demonstrated the best performance, followed closely by I-KAHAN. Conversely, for the GossipCop dataset, the SAFE model emerged as the best performer, succeeded by the SAME model, with I-KAHAN taking the third position.

Table 8.1: Comparison between the proposed I-KAHAN and various state-of-the-art multi-modal fake news detection models.

Model	F1 Score	
	PolitiFact	GossipCop
SAME [63]	0.7678	0.810
SAFE [11]	0.896	**0.895**
EANN [61]	0.7035	0.7123
FND-CLIP [13]	**0.9285**	0.783
I-KAHAN	0.901	0.8005

When interpreting these results, it is important to consider that the models were executed under different conditions. Factors such as varying hyperparameters and dataset versions can significantly influence the outcomes. For instance, according to the published research, the original KAHAN model achieved an F1 score of 0.9573 on the GossipCop dataset. However, the experimentation conducted in this thesis only yielded a score of 0.7892. This difference highlights how different experimental conditions can affect the model's performance. Moreover, the variation in performance could also be attributed to variations in the evaluation methods used in different studies. In this work, the scores were averaged as mentioned earlier, whereas other studies might consider the highest achieved score. When working with the KAHAN codebase, it was observed that it recorded only the highest score across the folds of the k-fold cross-validation, which might explain the discrepancy between the performance of KAHAN in this experiment and the results presented in the paper.

To ensure a fair and comprehensive comparison, an ideal approach could have been to implement and test all models, including I-KAHAN, on the same dataset or, alternatively, run I-KAHAN on the datasets used by the other models. However, this approach was impracticable due to the lack of publicly available code and datasets from many of the state-of-the-art model publications. Nonetheless, this comparison provides a general idea of how I-KAHAN aligns with the current state-of-the-art, acknowledging the noted limitations.

Apart from the potential differences resulting from varying datasets and experimental conditions, there could be intrinsic attributes to I-KAHAN that might have led to its comparatively lower performance. This includes the relative simplicity of I-KAHAN's integration techniques compared to the other models.

Several state-of-the-art models employ intricate fusion strategies and sophisticated methods to integrate multimodal information. For instance, models like FND-CLIP and SAFE leverage advanced techniques such as cross-modal attention mechanisms or hierarchical learning strategies to better capture the interplay between textual and visual information.

On the other hand, I-KAHAN, while efficient, is fundamentally simpler in its architecture. Its primary focus was on combining a range of proven techniques rather than exploring complex integration methods. This straightforward approach, while beneficial in terms of interpretability and ease of implementation, might not capture the intricacies of multi-modal information as effectively as the more sophisticated methods employed by other models.

8.4 Strengths and Limitations of I-KAHAN

This section navigates through the strengths and limitations of the proposed I-KAHAN fake news detection framework. It delves into its unique contributions, the inherent limitations of the system, and ethical implications.

8.4.1 Significance of I-KAHAN in the Domain of Fake News Detection

I-KAHAN positions itself in the field of fake news detection as a unique framework, most notably through its systematic approach to exploring alternative methods for image representation and integration with other features. It goes beyond simply creating a model; it offers invaluable insights into the comparative efficacy of different techniques, setting a helpful precedent for future research.

In its toolbox, I-KAHAN includes the novel IHAN method, with its creative use of attention mechanisms, exhibiting impressive performance. Furthermore, the incorporation of CLIP for image representation introduces an innovative twist to image embedding, by utilizing entity attention. While this application of attention did not consistently improve results, it illuminates the potential that attention-based techniques hold in enhancing fake news detection systems. The use of CLIP within the field has also only been briefly explored, and the addition of attention has to my knowledge never been explored before in previous work.

I-KAHAN also distinguishes itself through its comprehensive design. It utilizes three distinct news attributes: images, news text, and comments, and supplements these with external knowledge from a knowledge base. Although other such multi-modal systems exist, only a few utilize attention and external knowledge to this extent, especially not on image embeddings. This multi-dimensional approach has proven effective, outperforming the baseline by over 1% on the GossipCop dataset and 3% on the PolitiFact dataset.

8.4.2 Boundaries of I-KAHAN

While I-KAHAN has shown promise, it is also constrained by its limitations. The performance of I-KAHAN configurations diverges considerably between the GossipCop and PolitiFact datasets. This discrepancy suggests potential issues with the system's generalizability across different categories of news, a challenge that could be partially due to the limited scope of datasets considered in this work. This also reflects a broader issue within the field of fake news detection, where comprehensive, high-quality datasets are scarce.

Another critical concern worth mentioning is the discernible bias observable in I-KAHAN's classification outcomes. Detailed analysis of I-KAHAN's results reveals a propensity towards false positives when handling the PolitiFact dataset, notable for its higher concentration of fake news. In contrast, when processing the GossipCop dataset, which is characterized by a predominance of real news, I-KAHAN demonstrated a tendency towards false negatives. This dichotomy suggests that I-KAHAN struggles with effectively managing imbalanced datasets, an intricate problem discussed previously.

Additionally, when compared to other state-of-the-art models, as seen in the performance comparison section, the focus of I-KAHAN on combining a set of optimal techniques rather than inventing new fusion strategies might limit its potential to capture more nuanced patterns. This approach, while providing a more direct understanding of how each component contributes to the overall performance and offering clear pathways for further improvement, might slightly compromise the model's performance.

Notably, the current framework of I-KAHAN does not encompass all possible features relevant to fake news detection. The system focuses primarily on images, news text, and comments, in conjunction with external knowledge. However, other valuable information, such as user metadata is not utilized. The lack of additional contextual features poses a boundary to the capabilities of I-KAHAN, potentially limiting the richness of the model's understanding and the nuances in its detection abilities.

8.4.3 Ethical Implications

The real-world deployment of I-KAHAN, and fake news detection systems in general, brings forward several ethical considerations. Among these is the potential for false positives. If these systems were implemented in such a way that posts deemed by the system as fake news, would be blocked, there is a risk of infringing on freedom of speech. This concern is not limited to I-KAHAN but extends to all fake news detection systems, emphasizing the need for careful, ethically-informed implementations.

Similarly, there is a risk that the model might develop biases toward certain writing styles or images. This could inadvertently lead to instances where valid information is wrongly classified as fake news, potentially causing discrimination and unjust treatment.

Moreover, the possibility of exploitation by malicious actors is a real concern. For instance, they could mimic the styles or elements of credible news sources to enhance the perceived authenticity of their fake news. While I-KAHAN's multi-dimensional approach, which includes comments and news text, provides some level of protection, it underscores the need for continual development and refinement in fake news detection systems, always guided by ethical considerations.

8.5 Technical Challenges

Through working on this thesis several technical challenges were encountered, including integration into an existing codebase, the handling of complex programming challenges, as well as learning and optimizing the use of the IDUN cluster to conduct extensive experimentation.

8.5.1 Integration into an Existing Codebase

The process of integrating into an existing codebase represented a significant challenge. Understanding the code proved formidable due to the incompleteness of the accompanying documentation, which comprised only comments in the code, the KAHAN research paper, only scratching the surface on technical details, and a README file in the GitHub repository of KAHAN. In addition, unsuccessful attempts at communicating with the original authors left many questions unanswered.

Overcoming the challenge of identifying the appropriate data files necessary for embedding the textual content was crucial to recreate the same experimental setup as the baseline. Determining the file locations for the external knowledge embedding data necessary for embedding entities and claims was particularly challenging. This necessitated an exhaustive investigation and extensive web searches. To alleviate such difficulties in future research, the data sources used in this study have been clearly indicated, and downloadable links have been provided.

Integrating the necessary format for comments so that they could be processed by the existing comment-processing pipeline in the codebase also proved challenging. This required a comprehensive understanding of the codebase and substantial trial and error. It is worth noting that the original pipeline was not equipped to handle instances with no comments. However, such cases were included in this study due to the shortage of available data and the need to ensure that all instances contained textual content and an image. Adapting the pipeline to handle these instances necessitated significant modifications, adding an extra layer of complexity.

8.5.2 Managing Complex Programming Challenges

Complex programming tasks added another layer of difficulty to the study. To optimize the training process, image and text preprocessing tasks, including the embedding process and all other required preparation, were segregated into a separate script that was executed before the training loop. This division drastically reduced the training time, from several days to just a few hours. This was important due to the extensive amount of run-time needed to get the complete experimental results.

Incorporating various image integration components represented a demanding task due to their sheer number and the need to consider all alternative methods outlined in the experiments. This resulted in an extensive implementation phase that involved integrating multiple components, each with its unique characteristics and interactions. To accommodate this complexity, the script was modified to accept a comprehensive set of parameters, allowing the same script to be applied across all possible configurations simply by altering the input arguments. This undertaking required considerable debugging and presented significant complexity in its implementation. Nevertheless, it greatly simplified the process of conducting the experiments, turning a challenge into a streamlined process.

8.5.3 Optimizing the Utilization of the IDUN Cluster

Utilizing the IDUN cluster for experiments presented its own set of challenges. The learning curve was steep, requiring mastering the use of shell scripts and the command interface of Ubuntu to interact with the server where the scripts were run. This also entailed understanding the SLURM workload manager to execute, monitor, and manage jobs. Further, understanding the hardware of the IDUN cluster, the available resources, and their usage through shell script parameters was crucial.

The heavy demand for the IDUN cluster resources led to queuing issues and resource constraints. As detailed in Chapter 6, a total of 312 different runs were performed on the IDUN cluster to obtain the results presented in Chapter 7. Each of these runs required its own CPU core, placing a substantial demand on the cluster's resources. Discovering an efficient method to run these experiments required considerable trial and error. The final solution involved the use of array jobs and the generation of multiple array jobs from a complex loop. The configurations were defined in a separate text file for easy modification, resulting in a more efficient run-time.

CHAPTER

NINE

CONCLUSION AND FUTURE WORK

In this final chapter, we revisit the research questions outlined at the beginning of this thesis. The discussion aims to assess whether these questions have been satisfactorily addressed and to what extent the thesis objectives have been achieved. The chapter is divided into two sections: the first section provides a conclusive summary of the findings, and the second section identifies potential avenues for future work in the field of multi-modal fake news detection.

9.1 Conclusion and Research Contributions

This thesis delved into the domain of misinformation detection and proposed an enhancement to an existing detection model by incorporating the visual attributes of news. It is grounded in the historical progression from uni-modal to multi-modal fake news detection techniques. Building upon this foundation, the study introduced the Image-enhanced Knowledge-Aware Hierarchical Attention Network (I-KAHAN) model, which creatively combines textual and visual data to enhance the performance and robustness of fake news detection. The significant contributions of this research lie in the meticulous experimentation with various image integration techniques and the successful enhancement of the FakeNewsNet dataset, referred to as FakeNewsNet+. These efforts resulted in notable improvements, with some cases demonstrating improvements of up to 10% compared to FakeNewsNet.

The ensuing discussion delves into the specific findings of the study, addressing each research question (RQ) in turn:

RQ1 *Which techniques are most effective for incorporating visual elements into a multi-modal fake news detection system to enhance its classification performance?*

The research revealed the potential of various image integration techniques for improving fake news detection. CLIP-based image embeddings, pooling operations for dimensionality reduction, and concatenation for feature fusion emerged as the most effective techniques. Additionally, the study introduced a novel dimensionality reduction method, IHAN, and extensively explored the use of external knowledge and attention mechanisms to enhance the representation. Notably, the combination of CLIP with entity attention and the utilization of IHAN achieved remarkable performance. In addition, the study explored different classifier architectures and surprisingly found that a simpler neural architecture with only one hidden layer outperformed the one with an additional layer.

RQ2 *How significantly does the integration of visual attributes into a fake news detection system influence its classification performance?*

The significant improvement in the classification performance of the fake news detection system reaffirms the importance of integrating visual attributes. The I-KAHAN model outperformed the baseline, demonstrating an increase of over 1% in performance on the GossipCop dataset and 3% on the PolitiFact dataset. Since all other aspects of these systems remained the same, except for the integration of images in I-KAHAN, it suggests that the inclusion of visual aspects has a positive influence on classification performance.

RQ3 *How does the classification performance of the developed multi-modal fake news detection system, which includes visual elements, compare with existing state-of-the-art multi-modal systems?*

While the I-KAHAN model, incorporating visual elements, did not surpass all state-of-the-art multi-modal detection systems, it ranked second best on PolitiFact and third best on GossipCop among the four multi-modal systems it was compared to. This result is respectable, but not exceptional. Possible reasons for this discrepancy could be attributed to the relatively simplistic image integration approach of I-KAHAN compared to the more complex strategies employed by the other architectures. Another plausible reason could be that the performance of the other models was obtained from their respective papers and not re-evaluated under the same setup and with the same datasets as I-KAHAN. To strengthen this hypothesis, it is worth mentioning that the baseline KAHAN performed significantly worse during the experimentation conducted in this thesis compared to the results presented in the original paper. It is also possible that the evaluation methods employed by the other models differ, such as considering the highest achieved score during training as the final score, rather than the average score as in the case of this study.

9.2 Future Work

The promising results of this thesis not only validate the efficacy of multi-modal fake news detection but also illuminate various avenues for future research. It is important to approach these potential enhancements systematically and comprehensively, as they are interconnected and contribute collectively to the ongoing fight against misinformation.

One area that holds significant potential for advancement is the extended use of the CLIP encoder to embed textual attributes. By unifying the representation of visual and textual elements in the same vector space, this enhancement could improve the classifier's learning process. It would enable the model to discover nuanced correlations among diverse news attributes more effectively, leading to improved classification performance.

Improving the data collection process emerges as a critical facet for future research. The incorporation of sophisticated data scraping tools, such as the recent WorkGPT framework, could greatly enhance the quality of data collection. Additionally, expanding the range of data sources and employing synthetic data augmentation techniques could increase the volume of data available for training the model. This, in turn, would assist the model in learning and generalizing patterns more accurately, thereby enhancing its overall performance.

Addressing the challenge of dataset imbalance is also essential for improving the model's performance. Techniques like under-sampling, over-sampling, or SMOTE can help maintain a balance in the datasets, leading to more unbiased and robust results. Furthermore, diversifying the scope of news categories by incorporating additional datasets could enhance the model's generalizability across varied news topics and styles.

Considering the potential misuse of these models by malicious actors is a critical consideration. The ability of malefactors to mimic the styles of credible news sources to enhance the perceived authenticity of their fake news highlights the importance of continuous development and refinement of fake news detection systems. One possible direction is the inclusion of even more news attributes, building on the improved performance observed with the addition of images. Ethical considerations should always guide this approach, mitigating the risk and bolstering the model's resilience against such misuse.

Lastly, ethical considerations regarding freedom of speech warrant careful deliberation. While the necessity for efficient fake news detection is undeniable, it must be balanced with the preservation of this fundamental right. To achieve a delicate equilibrium, dedicated research is required to ensure that these technologies foster a healthy information ecosystem, where misinformation is combated without infringing upon freedom of speech.

In summary, the fight against misinformation is an ongoing endeavor, and each step toward improvement brings us closer to robust and effective models. This study has already illuminated promising pathways, reinforcing the importance and potential of multi-modal fake news detection. The suggested enhancements, intertwined with one another, collectively contribute to strengthening this line of research and making a meaningful impact in the battle against fake news.

BIBLIOGRAPHY

[1] Ø. L. Nilsen, *Multimodal Fake News Detection - Utilization of Images for Improved News Classification Performance*, Dec. 2022.

[2] A. Bondielli and F. Marcelloni, 'A survey on fake news and rumour detection techniques,' en, *Information Sciences*, vol. 497, pp. 38–55, Sep. 2019, issn: 0020-0255. doi: 10.1016/j.ins.2019.05.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519304372 (visited on 22/11/2022).

[3] H. Allcott and M. Gentzkow, 'Social Media and Fake News in the 2016 Election,' en, *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, May 2017, issn: 0895-3309. doi: 10.1257/jep.31.2.211. [Online]. Available: https://pubs.aeaweb.org/doi/10.1257/jep.31.2.211 (visited on 22/11/2022).

[4] M. Sallam, D. Dababseh, A. Yaseen, A. Al-Haidar, D. Taim, H. Eid, N. A. Ababneh, F. G. Bakri and A. Mahafzah, 'COVID-19 misinformation: Mere harmless delusions or much more? A knowledge and attitude cross-sectional study among the general public residing in Jordan,' en, *PLOS ONE*, vol. 15, no. 12, e0243264, Dec. 2020, Publisher: Public Library of Science, issn: 1932-6203. doi: 10.1371/journal.pone.0243264. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0243264 (visited on 22/05/2023).

[5] Y. M. Rocha, G. A. de Moura, G. A. Desidério, C. H. de Oliveira, F. D. Lourenço and L. D. de Figueiredo Nicolete, 'The impact of fake news on social media and its influence on health during the COVID-19 pandemic: A systematic review,' en, *Journal of Public Health*, Oct. 2021, issn: 1613-2238. doi: 10.1007/s10389-021-01658-z. [Online]. Available: https://doi.org/10.1007/s10389-021-01658-z (visited on 22/05/2023).

[6] Y. Shin, Y. Sojdehei, L. Zheng and B. Blanchard, 'Content-Based Unsupervised Fake News Detection on Ukraine-Russia War,' en, vol. 7, no. 1,

[7] L. De Angelis, F. Baglivo, G. Arzilli, G. P. Privitera, P. Ferragina, A. E. Tozzi and C. Rizzo, 'ChatGPT and the rise of large language models: The new AI-driven infodemic threat in public health,' *Frontiers in Public Health*, vol. 11, p. 1 166 120,

Apr. 2023, issn: 2296-2565. doi: 10.3389/fpubh.2023.1166120. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10166793/ (visited on 12/06/2023).

[8] K. Shu, A. Sliva, S. Wang, J. Tang and H. Liu, *Fake News Detection on Social Media: A Data Mining Perspective*, arXiv:1708.01967 [cs], Sep. 2017. [Online]. Available: http://arxiv.org/abs/1708.01967 (visited on 22/11/2022).

[9] Y.-W. Tseng, H.-K. Yang, W.-Y. Wang and W.-C. Peng, 'KAHAN: Knowledge-Aware Hierarchical Attention Network for Fake News detection on Social Media,' en, in *Companion Proceedings of the Web Conference 2022*, Virtual Event, Lyon France: ACM, Apr. 2022, pp. 868–875, isbn: 978-1-4503-9130-6. doi: 10.1145/3487553.3524664. [Online]. Available: https://dl.acm.org/doi/10.1145/3487553.3524664 (visited on 16/05/2023).

[10] N. Ahuja and S. Kumar, 'Fusion of Semantic, Visual and Network Information for Detection of Misinformation on Social Media,' *Cybernetics and Systems*, vol. 0, no. 0, pp. 1–23, Oct. 2022, Publisher: Taylor & Francis, issn: 0196-9722. doi: 10.1080/01969722.2022.2130248. [Online]. Available: https://doi.org/10.1080/01969722.2022.2130248 (visited on 01/03/2023).

[11] X. Zhou, J. Wu and R. Zafarani, *SAFE: Similarity-Aware Multi-Modal Fake News Detection*, arXiv:2003.04981 [cs, stat], Feb. 2020. [Online]. Available: http://arxiv.org/abs/2003.04981 (visited on 21/04/2023).

[12] J. Cao, P. Qi, Q. Sheng, T. Yang, J. Guo and J. Li, 'Exploring the Role of Visual Content in Fake News Detection,' en, in *Disinformation, Misinformation, and Fake News in Social Media: Emerging Research Challenges and Opportunities*, ser. Lecture Notes in Social Networks, K. Shu, S. Wang, D. Lee and H. Liu, Eds., Cham: Springer International Publishing, 2020, pp. 141–161, isbn: 978-3-030-42699-6. doi: 10.1007/978-3-030-42699-6_8. [Online]. Available: https://doi.org/10.1007/978-3-030-42699-6_8 (visited on 23/05/2023).

[13] Y. Zhou, Q. Ying, Z. Qian, S. Li and X. Zhang, *Multimodal Fake News Detection via CLIP-Guided Learning*, arXiv:2205.14304 [cs], May 2022. [Online]. Available: http://arxiv.org/abs/2205.14304 (visited on 18/05/2023).

[14] I. Segura-Bedmar and S. Alonso-Bartolome, 'Multimodal Fake News Detection,' en, *Information*, vol. 13, no. 6, p. 284, Jun. 2022, Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, issn: 2078-2489. doi: 10.3390/info13060284. [Online]. Available: https://www.mdpi.com/2078-2489/13/6/284 (visited on 12/06/2023).

[15] E. Mustafaraj and P. T. Metaxas, *The Fake News Spreading Plague: Was it Preventable?* arXiv:1703.06988 [cs], Mar. 2017. doi: 10.48550/arXiv.1703.06988. [Online]. Available: http://arxiv.org/abs/1703.06988 (visited on 22/05/2023).

[16] C. A. Watson, ‘Information Literacy in a Fake/False News World: An Overview of the Characteristics of Fake News and its Historical Development,’ en, *International Journal of Legal Information*, vol. 46, no. 2, pp. 93–96, Jul. 2018, Publisher: Cambridge University Press, issn: 0731-1265, 2331-4117. doi: 10.1017/jli.2018.25. [Online]. Available: https://www.cambridge.org/core/journals/international-journal-of-legal-information/article/information-literacy-in-a-fakefalse-news-world-an-overview-of-the-characteristics-of-fake-news-and-its-historical-development/674B5DCB6A3106FC50B5DE1CCF0646AA (visited on 23/05/2023).

[17] P. Hernon, ‘Disinformation and misinformation through the internet: Findings of an exploratory study,’ en, *Government Information Quarterly*, vol. 12, no. 2, pp. 133–139, Jan. 1995, issn: 0740-624X. doi: 10.1016/0740-624X(95)90052-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0740624X95900527 (visited on 22/05/2023).

[18] F. Monti, F. Frasca, D. Eynard, D. Mannion and M. M. Bronstein, *Fake News Detection on Social Media using Geometric Deep Learning*, arXiv:1902.06673 [cs, stat], Feb. 2019. [Online]. Available: http://arxiv.org/abs/1902.06673 (visited on 23/05/2023).

[19] G. Rebala, A. Ravi and S. Churiwala, ‘Machine Learning Definition and Basics,’ en, in *An Introduction to Machine Learning*, G. Rebala, A. Ravi and S. Churiwala, Eds., Cham: Springer International Publishing, 2019, pp. 1–17, isbn: 978-3-030-15729-6. doi: 10.1007/978-3-030-15729-6_1. [Online]. Available: https://doi.org/10.1007/978-3-030-15729-6_1 (visited on 23/05/2023).

[20] Y. LeCun, Y. Bengio and G. Hinton, ‘Deep learning,’ en, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, Number: 7553 Publisher: Nature Publishing Group, issn: 1476-4687. doi: 10.1038/nature14539. [Online]. Available: https://www.nature.com/articles/nature14539 (visited on 19/05/2023).

[21] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain and A. J. Aljaaf, ‘A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science,’ en, in *Supervised and Unsupervised Learning for Data Science*, ser. Unsupervised and Semi-Supervised Learning, M. W. Berry, A. Mohamed and B. W. Yap, Eds., Cham: Springer International Publishing, 2020, pp. 3–21, isbn: 978-3-030-22475-2. doi: 10.1007/978-3-030-22475-2_1. [Online]. Available: https://doi.org/10.1007/978-3-030-22475-2_1 (visited on 23/05/2023).

[22] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun and M. Goldblum, *A Cookbook of Self-Supervised Learning*, arXiv:2304.12210 [cs], Apr. 2023. [Online]. Available: http://arxiv.org/abs/2304.12210 (visited on 19/06/2023).

[23] Z. Ghahramani, ‘Unsupervised Learning,’ en, in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, ser. Lecture Notes in Computer Science, O. Bousquet, U. von Luxburg and G. Rätsch, Eds., Berlin, Heidelberg: Springer, 2004, pp. 72–112, isbn: 978-3-540-28650-9. doi: 10.1007/978-3-540-28650-9_5. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_5 (visited on 24/05/2023).

[24] L. E. Peterson, ‘K-nearest neighbor,’ en, *Scholarpedia*, vol. 4, no. 2, p. 1883, Feb. 2009, issn: 1941-6016. doi: 10.4249/scholarpedia.1883. [Online]. Available: http://www.scholarpedia.org/article/K-nearest_neighbor (visited on 18/06/2023).

[25] F. Rosenblatt, ‘The perceptron: A probabilistic model for information storage and organization in the brain.,’ en, *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, issn: 1939-1471, 0033-295X. doi: 10.1037/h0042519. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0042519 (visited on 19/05/2023).

[26] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*, en. The MIT Press, 2017, isbn: 978-0-262-34393-0. doi: 10.7551/mitpress/11301.001.0001. [Online]. Available: https://direct.mit.edu/books/book/3132/perceptronsan-introduction-to-computational (visited on 24/05/2023).

[27] A. Jain, J. Mao and K. Mohiuddin, ‘Artificial neural networks: A tutorial,’ *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996, Conference Name: Computer, issn: 1558-0814. doi: 10.1109/2.485891.

[28] M. W. Gardner and S. R. Dorling, ‘Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,’ en, *Atmospheric Environment*, vol. 32, no. 14, pp. 2627–2636, Aug. 1998, issn: 1352-2310. doi: 10.1016/S1352-2310(97)00447-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1352231097004470 (visited on 24/05/2023).

[29] D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D. Scherer, A. Müller and S. Behnke, ‘Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition,’ en, in *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch and L. S. Iliadis, Eds., vol. 6354, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101, isbn: 978-3-642-15824-7 978-3-642-15825-4. doi: 10.1007/978-3-642-15825-4_10. [Online]. Available: http://link.springer.com/10.1007/978-3-642-15825-4_10 (visited on 19/05/2023).

[30] *Understanding LSTM Networks – colah's blog.* [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 16/06/2023).

[31] M. Schuster and K. Paliwal, 'Bidirectional recurrent neural networks,' *Signal Processing, IEEE Transactions on*, vol. 45, pp. 2673–2681, Dec. 1997. doi: `10.1109/78.650093`.

[32] W. James, *The Principles of Psychology, Vol. 1*, English, Revised ed. edition. New York: Dover Publications, Jun. 1950, isbn: 978-0-486-20381-2.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, *Attention Is All You Need*, arXiv:1706.03762 [cs], Dec. 2017. [Online]. Available: `http://arxiv.org/abs/1706.03762` (visited on 19/05/2023).

[34] J. Lu, J. Yang, D. Batra and D. Parikh, *Hierarchical Question-Image Co-Attention for Visual Question Answering*, arXiv:1606.00061 [cs], Jan. 2017. [Online]. Available: `http://arxiv.org/abs/1606.00061` (visited on 19/05/2023).

[35] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola and E. Hovy, 'Hierarchical Attention Networks for Document Classification,' en, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, 2016, pp. 1480–1489. doi: `10.18653/v1/N16-1174`. [Online]. Available: `http://aclweb.org/anthology/N16-1174` (visited on 02/10/2022).

[36] Daniel Jurafsky and James H. Martin, *Speech and Language Processing, 3rd Edition*, 3rd (draft). Jan. 2023. [Online]. Available: `https://web.stanford.edu/~jurafsky/slp3/ed3book_jan72023.pdf` (visited on 20/06/2023).

[37] Z. S. Harris, 'Distributional Structure,' en, *WORD*, vol. 10, no. 2-3, pp. 146–162, Aug. 1954, issn: 0043-7956, 2373-5112. doi: `10.1080/00437956.1954.11659520`. [Online]. Available: `http://www.tandfonline.com/doi/full/10.1080/00437956.1954.11659520` (visited on 19/05/2023).

[38] T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, arXiv:1301.3781 [cs], Sep. 2013. [Online]. Available: `http://arxiv.org/abs/1301.3781` (visited on 19/05/2023).

[39] J. Pennington, R. Socher and C. Manning, 'GloVe: Global Vectors for Word Representation,' in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. doi: `10.3115/v1/D14-1162`. [Online]. Available: `https://aclanthology.org/D14-1162` (visited on 17/06/2023).

[40] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805 [cs], May 2019. [Online]. Available: `http://arxiv.org/abs/1810.04805` (visited on 17/06/2023).

[41] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv:1409.1556 [cs], Apr. 2015. [Online]. Available: `http://arxiv.org/abs/1409.1556` (visited on 23/11/2022).

[42] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385 [cs], Dec. 2015. [Online]. Available: `http://arxiv.org/abs/1512.03385` (visited on 19/05/2023).

[43] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, 'Going deeper with convolutions,' en, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 1–9, isbn: 978-1-4673-6964-0. doi: `10.1109/CVPR.2015.7298594`. [Online]. Available: `http://ieeexplore.ieee.org/document/7298594/` (visited on 17/06/2023).

[44] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, M. Tomizuka, K. Keutzer and P. Vajda, *Visual Transformers: Token-based Image Representation and Processing for Computer Vision*. Jun. 2020.

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, *Learning Transferable Visual Models From Natural Language Supervision*, arXiv:2103.00020 [cs], Feb. 2021. [Online]. Available: `http://arxiv.org/abs/2103.00020` (visited on 19/05/2023).

[46] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin and I. Misra, *ImageBind: One Embedding Space To Bind Them All*, arXiv:2305.05665 [cs], May 2023. [Online]. Available: `http://arxiv.org/abs/2305.05665` (visited on 17/06/2023).

[47] H. Narasimhan, W. Pan, P. Kar, P. Protopapas and H. G. Ramaswamy, 'Optimizing the Multiclass F-Measure via Biconcave Programming,' en, in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Barcelona, Spain: IEEE, Dec. 2016, pp. 1101–1106, isbn: 978-1-5090-5473-2. doi: `10.1109/ICDM.2016.0143`. [Online]. Available: `http://ieeexplore.ieee.org/document/7837956/` (visited on 18/06/2023).

[48] E. A. Nismi Mol and M. B. Santosh Kumar, 'Review on knowledge extraction from text and scope in agriculture domain,' en, *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4403–4445, May 2023, issn: 1573-7462. doi: `10.1007/s10462-022-10239-9`. [Online]. Available: `https://doi.org/10.1007/s10462-022-10239-9` (visited on 19/06/2023).

[49] Y. Dun, K. Tu, C. Chen, C. Hou and X. Yuan, 'KAN: Knowledge-aware Attention Network for Fake News Detection,' en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 81–89, May 2021, Number: 1, issn: 2374-3468. doi: `10.1609/aaai.v35i1.16080`. [Online]. Available: `https://ojs.aaai.org/index.php/AAAI/article/view/16080` (visited on 04/05/2023).

[50] L. Hu, T. Yang, L. Zhang, W. Zhong, D. Tang, C. Shi, N. Duan and M. Zhou, 'Compare to The Knowledge: Graph Neural Fake News Detection with External Knowledge,' in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Lan-*

guage Processing (Volume 1: Long Papers), Online: Association for Computational Linguistics, Aug. 2021, pp. 754–763. doi: 10.18653/v1/2021.acl-long.62. [Online]. Available: https://aclanthology.org/2021.acl-long.62 (visited on 20/05/2023).

[51] R. J. Wilson, *Introduction to graph theory*, en, 4. ed., [Nachdr.] Harlow Munich: Prentice Hall, 2009, isbn: 978-0-582-24993-6.

[52] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui and P. S. Yu, 'Heterogeneous Graph Attention Network,' en, in *The World Wide Web Conference*, San Francisco CA USA: ACM, May 2019, pp. 2022–2032, isbn: 978-1-4503-6674-8. doi: 10.1145/3308558.3313562. [Online]. Available: https://dl.acm.org/doi/10.1145/3308558.3313562 (visited on 18/06/2023).

[53] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach* (Pearson series in artificial intelligence), eng, Fourth edition. Hoboken, NJ: Pearson, 2021, OCLC: 1124776132, isbn: 978-0-13-461099-3.

[54] L. Ehrlinger and W. Wöß, 'Towards a Definition of Knowledge Graphs,' Sep. 2016.

[55] S. Ji, S. Pan, E. Cambria, P. Marttinen and P. S. Yu, 'A Survey on Knowledge Graphs: Representation, Acquisition, and Applications,' *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, Feb. 2022, Conference Name: IEEE Transactions on Neural Networks and Learning Systems, issn: 2162-2388. doi: 10.1109/TNNLS.2021.3070843.

[56] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, 'Freebase: A collaboratively created graph database for structuring human knowledge,' en, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, Vancouver Canada: ACM, Jun. 2008, pp. 1247–1250, isbn: 978-1-60558-102-6. doi: 10.1145/1376616.1376746. [Online]. Available: https://dl.acm.org/doi/10.1145/1376616.1376746 (visited on 20/05/2023).

[57] T. Pellissier Tanon, G. Weikum and F. Suchanek, 'YAGO 4: A Reason-able Knowledge Base,' en, in *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase and M. Cochez, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 583–596, isbn: 978-3-030-49461-2. doi: 10.1007/978-3-030-49461-2_34.

[58] D. Vrandečić and M. Krötzsch, 'Wikidata: A free collaborative knowledgebase,' en, *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014, issn: 0001-0782, 1557-7317. doi: 10.1145/2629489. [Online]. Available: https://dl.acm.org/doi/10.1145/2629489 (visited on 20/05/2023).

[59] I. Yamada, A. Asai, J. Sakuma, H. Shindo, H. Takeda, Y. Takefuji and Y. Matsumoto, 'Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia,' in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, Oct. 2020, pp. 23–30.

doi: 10.18653/v1/2020.emnlp-demos.4. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.4 (visited on 19/05/2023).

[60] K. Shu, L. Cui, S. Wang, D. Lee and H. Liu, 'dEFEND: Explainable Fake News Detection,' in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 395–405, isbn: 978-1-4503-6201-6. doi: 10.1145/3292500.3330935. [Online]. Available: https://dl.acm.org/doi/10.1145/3292500.3330935 (visited on 04/05/2023).

[61] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su and J. Gao, '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*, ser. KDD '18, New York, NY, USA: Association for Computing Machinery, Jul. 2018, pp. 849–857, isbn: 978-1-4503-5552-0. doi: 10.1145/3219819.3219903. [Online]. Available: https://doi.org/10.1145/3219819.3219903 (visited on 01/03/2023).

[62] R. K. Kaliyar, A. Goswami and P. Narang, 'FakeBERT: Fake news detection in social media with a BERT-based deep learning approach,' en, *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 11 765–11 788, Mar. 2021, issn: 1573-7721. doi: 10.1007/s11042-020-10183-2. [Online]. Available: https://doi.org/10.1007/s11042-020-10183-2 (visited on 18/05/2023).

[63] L. Cui, S. Wang and D. Lee, 'SAME: Sentiment-aware multi-modal embedding for detecting fake news,' en, in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Vancouver British Columbia Canada: ACM, Aug. 2019, pp. 41–48, isbn: 978-1-4503-6868-1. doi: 10.1145/3341161.3342894. [Online]. Available: https://dl.acm.org/doi/10.1145/3341161.3342894 (visited on 16/05/2023).

[64] K. Shu, D. Mahudeswaran, S. Wang, D. Lee and H. Liu, *FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media*, arXiv:1809.01286 [cs], Mar. 2019. doi: 10.48550/arXiv.1809.01286. [Online]. Available: http://arxiv.org/abs/1809.01286 (visited on 20/04/2023).

[65] *Using The Wayback Machine – Internet Archive Help Center*, en-US. [Online]. Available: https://help.archive.org/help/using-the-wayback-machine/ (visited on 15/05/2023).

[66] P. Liashchynskyi and P. Liashchynskyi, *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*, arXiv:1912.06059 [cs, stat], Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.06059 (visited on 25/06/2023).

[67] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, 'SMOTE: Synthetic Minority Over-sampling Technique,' *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, arXiv:1106.1813 [cs], issn: 1076-9757. doi:

10.1613/jair.953. [Online]. Available: http://arxiv.org/abs/1106.1813 (visited on 25/06/2023).

Appendices

APPENDIX

ONE

EXTENDED EXPERIMENTAL RESULTS

This section contains a detailed presentation of the experimental results. To provide a comprehensive overview, Table A.1 presents the complete output of all conducted experiments, using the F1 score as the sole metric for simplicity. Additionally, for Experiment 1D, extended results for the PolitiFact and GossipCop datasets can be found in Table A.2 and Table A.3 respectively. Similarly, Table A.4 and Table A.5 present the extended results of Experiment 2 for each dataset. These tables encompass all configurations, surpassing the previously mentioned limit of ten configurations in Chapter 7.

Table A.1: F1 scores for different I-KAHAN architecture configurations. Each cell contains two scores: shallow (left) and deep (right). The best numbers in each column are shown in bold, while the second-best are underlined.

Configuration	GossipCop		PolitiFact	
	FakeNewsNet	FakeNewsNet+	FakeNewsNet	FakeNewsNet+
CLIP(EA)-Cat	0.7667/0.7487	0.7649/0.7487	0.8301/0.8175	0.8323/0.8175
CLIP-Cat	**0.8042**/0.7984	0.8005/0.7984	**0.9088**/0.8883	**0.9049**/0.8883
Resnet50-AvgPool-Avg	0.7929/0.7971	0.7929/0.7971	0.8702/0.8986	0.8702/0.8986
Resnet50-AvgPool-Cat	0.7898/0.7983	0.7975/0.7983	0.8867/0.8987	0.8867/0.8987
Resnet50-AvgPool-ElemMult	0.5743/0.5861	0.5887/0.5861	0.659/0.7857	0.6566/0.7857
Resnet50-DNN-Avg	0.7743/0.7878	**0.8235**/0.7878	0.8633/0.8464	0.8611/0.8464
Resnet50-DNN-Cat	0.7901/0.7913	0.8068/0.7913	0.8726/0.8427	0.8742/0.8427
Resnet50-DNN-ElemMult	0.4966/0.5809	0.4962/0.5809	0.6374/0.6571	0.6374/0.6571
Resnet50-FC-Avg	0.6842/0.678	0.6855/0.678	0.784/0.7827	0.784/0.7827
Resnet50-FC-Cat	0.6735/0.6668	0.6744/0.6668	0.7887/0.7816	0.7887/0.7816
Resnet50-FC-ElemMult	0.5331/0.5726	0.5374/0.5726	0.6175/0.7177	0.6175/0.7177
Resnet50-IHAN(EA)-Avg	0.7916/0.7893	0.79/0.7893	0.8617/0.8804	0.8617/0.8804
Resnet50-IHAN(EA)-Cat	0.7924/0.7917	0.7838/0.7917	0.8864/0.883	0.8864/0.883
Resnet50-IHAN(EA)-ElemMult	0.5773/0.5846	0.5696/0.5846	0.588/0.8091	0.6239/0.8091
Resnet50-IHAN-Avg	0.7845/0.7929	0.7949/0.7929	0.8702/0.8875	0.8702/0.8875
Resnet50-IHAN-Cat	0.7955/0.7921	0.7957/0.7921	0.885/0.892	0.8845/0.892
Resnet50-IHAN-ElemMult	0.5732/0.5856	0.5791/0.5856	0.6521/0.8022	0.6521/0.8022
Resnet50-MaxPool-Avg	0.7544/0.7597	0.7621/0.7597	0.8073/0.8647	0.8073/0.8647
Resnet50-MaxPool-Cat	0.7795/0.7927	0.7869/0.7927	0.8779/0.8742	0.8779/0.8742
Resnet50-MaxPool-ElemMult	0.581/0.5855	0.5823/0.5855	0.7447/0.7827	0.7447/0.7827
VGG19-AvgPool-Avg	0.7917/0.7989	0.7917/0.7989	0.794/0.9044	0.8746/0.9044
VGG19-AvgPool-Cat	0.7975/0.7937	0.7975/0.7937	0.8575/0.8985	0.8908/0.8985
VGG19-AvgPool-ElemMult	0.5766/0.5875	0.5766/0.5875	0.6438/0.8042	0.6892/0.8042
VGG19-DNN-Avg	0.7651/0.7951	0.7649/0.7951	0.8311/0.8623	0.8619/0.8623
VGG19-DNN-Cat	0.7691/0.7925	0.7691/0.7925	0.833/0.8698	0.8802/0.8698
VGG19-DNN-ElemMult	0.4956/0.5755	0.4956/0.5755	0.5984/0.7715	0.6532/0.7715
VGG19-FC-Avg	0.7129/0.7103	0.733/0.7103	0.7876/0.7916	0.7876/0.7916
VGG19-FC-Cat	0.7053/0.7	0.7053/0.7	0.7844/0.7912	0.7844/0.7912
VGG19-FC-ElemMult	0.5256/0.5726	0.5257/0.5726	0.6025/0.7402	0.6025/0.7402
VGG19-IHAN(EA)-Avg	0.793/0.7893	0.7931/0.7893	0.8585/0.8837	0.8585/0.8837
VGG19-IHAN(EA)-Cat	0.7925/0.7904	0.7941/0.7904	0.8791/0.8773	0.8791/0.8773
VGG19-IHAN(EA)-ElemMult	0.5654/0.5864	0.5696/0.5864	0.62/0.8095	0.62/0.8095
VGG19-IHAN-Avg	0.7883/0.7924	0.7899/0.7924	0.8718/0.8927	0.8718/0.8927
VGG19-IHAN-Cat	0.797/0.7928	0.8071/0.7928	0.876/0.8979	0.876/0.8979
VGG19-IHAN-ElemMult	0.572/0.5884	0.5882/0.5884	0.6546/0.795	0.6546/0.795
VGG19-MaxPool-Avg	0.7805/0.7793	0.7805/0.7793	0.8101/0.872	0.8737/0.872
VGG19-MaxPool-Cat	0.7925/0.7986	0.8031/0.7986	0.8918/0.8951	0.8918/0.8951
VGG19-MaxPool-ElemMult	0.5809/0.5827	0.5809/0.5827	0.6601/0.796	0.6955/0.796

Table A.2: Detailed comparison of the shallow and deep classifier on the PolitiFact dataset for all the configurations of I-KAHAN. The numbers on the left side of each slash are those of the shallow classifier, while those on the right-hand side are of the deep one. The highest numbers are in bold.

Configuration	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP(EA)-Cat	**0.8186**/0.8184	**0.8231**/0.8178	0.8125/**0.8188**	0.809/**0.8175**
CLIP-Cat	**0.9020**/0.8892	**0.9059**/0.8911	**0.8983**/0.8874	**0.901**/0.8883
Resnet50-AvgPool-Avg	0.8713/**0.8994**	0.8728/**0.8993**	0.8691/**0.8979**	0.870/**0.8986**
Resnet50-AvgPool-Cat	0.8875/**0.8995**	0.8873/**0.8988**	0.8863/**0.8980**	0.887/**0.8987**
Resnet50-AvgPool-ElemMult	0.6735/**0.7920**	0.6636/**0.8133**	0.6729/**0.7747**	0.659/**0.7857**
Resnet50-DNN-Avg	**0.8611**/0.8482	**0.8674**/0.8572	**0.8541**/0.8455	**0.859**/0.8464
Resnet50-DNN-Cat	**0.8653**/0.8448	**0.8677**/0.8564	**0.8684**/0.8426	**0.865**/0.8427
Resnet50-DNN-ElemMult	0.6498/**0.6931**	0.6913/**0.7553**	0.6546/**0.6558**	0.637/**0.6571**
Resnet50-FC-Avg	**0.7852**/0.7835	0.7844/**0.7850**	0.7846/**0.7867**	**0.784**/0.7827
Resnet50-FC-Cat	**0.7894**/0.7826	**0.7897**/0.7864	**0.7921**/0.7856	**0.789**/0.7816
Resnet50-FC-ElemMult	0.6303/**0.7246**	0.6815/**0.7277**	0.6471/**0.7096**	0.618/**0.7177**
Resnet50-IHAN(EA)-Avg	0.8628/**0.8815**	0.8626/**0.8821**	0.8607/**0.8782**	0.862/**0.8804**
Resnet50-IHAN(EA)-Cat	**0.8876**/0.8841	**0.8911**/0.8842	**0.8849**/0.8809	**0.886**/0.8830
Resnet50-IHAN(EA)-ElemMult	0.6387/**0.8150**	0.5826/**0.8397**	0.6595/**0.7981**	0.613/**0.8091**
Resnet50-IHAN-Avg	0.8713/**0.8883**	0.8709/**0.8885**	0.8685/**0.8869**	0.870/**0.8875**
Resnet50-IHAN-Cat	0.8850/**0.8926**	0.8859/**0.8922**	0.8836/**0.8923**	0.884/**0.8920**
Resnet50-IHAN-ElemMult	0.6827/**0.8082**	0.6141/**0.8311**	0.6781/**0.7911**	0.652/**0.8022**
Resnet50-MaxPool-Avg	0.8090/**0.8662**	0.8093/**0.8680**	0.8049/**0.8622**	0.807/**0.8647**
Resnet50-MaxPool-Cat	**0.8790**/0.8756	**0.8796**/0.8777	**0.8770**/0.8715	**0.878**/0.8742
Resnet50-MaxPool-ElemMult	0.7519/**0.7894**	0.7900/**0.8134**	0.7465/**0.7718**	0.745/**0.7827**
VGG19-AvgPool-Avg	0.8756/**0.9054**	0.8749/**0.9070**	0.8736/**0.9025**	0.875/**0.9044**
VGG19-AvgPool-Cat	0.8918/**0.8994**	0.8922/**0.8997**	0.8888/**0.8966**	0.891/**0.8985**
VGG19-AvgPool-ElemMult	0.7016/**0.8108**	0.6977/**0.8359**	0.7028/**0.7926**	0.689/**0.8042**
VGG19-DNN-Avg	0.8645/0.8645	**0.8733**/0.8687	0.8557/**0.8577**	0.862/**0.8623**
VGG19-DNN-Cat	**0.8816**/0.8721	**0.8854**/0.8809	**0.8772**/0.8631	**0.880**/0.8698
VGG19-DNN-ElemMult	0.6751/**0.7835**	0.7219/**0.8351**	0.6522/**0.7583**	0.653/**0.7715**
VGG19-FC-Avg	0.7895/**0.7929**	0.7891/**0.7926**	0.7859/**0.7925**	0.788/**0.7916**
VGG19-FC-Cat	0.7861/**0.7928**	0.7840/**0.7926**	0.7835/**0.7910**	0.784/**0.7912**
VGG19-FC-ElemMult	0.6147/**0.7468**	0.6734/**0.7540**	0.6369/**0.7317**	0.603/**0.7402**
VGG19-IHAN(EA)-Avg	0.8594/**0.8849**	0.8586/**0.8865**	0.8579/**0.8815**	0.858/**0.8837**
VGG19-IHAN(EA)-Cat	**0.8799**/0.8781	**0.8798**/0.8771	**0.8793**/0.8777	**0.879**/0.8773
VGG19-IHAN(EA)-ElemMult	0.6463/**0.8150**	0.5938/**0.8357**	0.6640/**0.7985**	0.620/**0.8095**
VGG19-IHAN-Avg	0.8730/**0.8935**	0.8729/**0.8929**	0.8701/**0.8923**	0.872/**0.8927**
VGG19-IHAN-Cat	0.8773/**0.8986**	0.8789/**0.8976**	0.8734/**0.8983**	0.876/**0.8979**
VGG19-IHAN-ElemMult	0.6844/**0.8014**	0.6162/**0.8259**	0.6807/**0.7834**	0.655/**0.7950**
VGG19-MaxPool-Avg	**0.8747**/0.8730	**0.8731**/0.8719	**0.8726**/0.8713	**0.874**/0.8720
VGG19-MaxPool-Cat	0.8926/**0.8960**	0.8924/**0.8968**	0.8911/**0.8940**	0.892/**0.8951**
VGG19-MaxPool-ElemMult	0.7024/**0.8031**	0.7425/**0.8295**	0.7077/**0.7841**	0.696/**0.7960**

Table A.3: Detailed comparison of the shallow and deep classifier on the GossipCop dataset for all the configurations of I-KAHAN. The numbers on the left side of each slash are those of the shallow classifier, while those on the right-hand side are of the deep one. The highest numbers are in bold.

Configuration	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP(EA)-Cat	**0.7657**/0.7494	**0.7669**/0.7499	**0.7650**/0.7484	**0.765**/0.7487
CLIP-Cat	**0.8011**/0.7987	**0.8020**/0.7986	**0.8001**/0.7991	**0.800**/0.7984
Resnet50-AvgPool-Avg	0.7932/**0.7973**	0.7934/**0.7973**	0.7935/**0.7979**	0.793/**0.7971**
Resnet50-AvgPool-Cat	0.7978/**0.7984**	0.7973/**0.7999**	0.7980/**0.8005**	0.798/**0.7983**
Resnet50-AvgPool-ElemMult	**0.6025**/0.6000	**0.6803**/0.6767	**0.6214**/0.6188	**0.589**/0.5861
Resnet50-DNN-Avg	**0.8242**/0.7881	**0.8247**/0.7919	**0.8222**/0.7898	**0.824**/0.7878
Resnet50-DNN-Cat	**0.8073**/0.7920	**0.8075**/0.7950	**0.8069**/0.7916	**0.807**/0.7913
Resnet50-DNN-ElemMult	0.5505/**0.5953**	0.4541/**0.6711**	0.5388/**0.6141**	0.496/**0.5809**
Resnet50-FC-Avg	**0.6867**/0.6786	**0.6858**/0.6787	**0.6841**/0.6781	**0.685**/0.6780
Resnet50-FC-Cat	**0.6757**/0.6677	**0.6767**/0.6680	**0.6740**/0.6668	**0.674**/0.6668
Resnet50-FC-ElemMult	0.5671/**0.5810**	0.5650/**0.6212**	0.5676/**0.5960**	0.537/**0.5726**
Resnet50-IHAN(EA)-Avg	**0.7901**/0.7894	0.7918/**0.7924**	0.7920/**0.7922**	**0.790**/0.7893
Resnet50-IHAN(EA)-Cat	0.7840/**0.7918**	0.7853/**0.7931**	0.7852/**0.7937**	0.784/**0.7917**
Resnet50-IHAN(EA)-ElemMult	0.5911/**0.5979**	0.6561/**0.6691**	0.5999/**0.6163**	0.570/**0.5846**
Resnet50-IHAN-Avg	**0.7950**/0.7930	**0.7955**/0.7940	**0.7964**/0.7945	**0.795**/0.7929
Resnet50-IHAN-Cat	**0.7959**/0.7924	**0.7965**/0.7937	**0.7968**/0.7935	**0.796**/0.7921
Resnet50-IHAN-ElemMult	0.5962/**0.6009**	0.6639/**0.6861**	0.6099/**0.6205**	0.579/**0.5856**
Resnet50-MaxPool-Avg	**0.7625**/0.7600	**0.7618**/0.7596	**0.7621**/0.7602	**0.762**/0.7597
Resnet50-MaxPool-Cat	0.7872/**0.7930**	0.7877/**0.7932**	0.7877/**0.7936**	0.787/**0.7927**
Resnet50-MaxPool-ElemMult	**0.5998**/0.5987	0.6702/**0.6708**	0.6133/**0.6171**	0.582/**0.5855**
VGG19-AvgPool-Avg	0.7920/**0.7991**	0.7923/**0.7995**	0.7925/**0.8002**	0.792/**0.7989**
VGG19-AvgPool-Cat	**0.7977**/0.7938	**0.7978**/0.7938	**0.7984**/0.7948	**0.797**/0.7937
VGG19-AvgPool-ElemMult	0.5944/**0.6016**	0.6627/**0.6804**	0.6079/**0.6206**	0.577/**0.5875**
VGG19-DNN-Avg	0.7667/**0.7955**	0.7709/**0.7982**	0.7634/**0.7967**	0.765/**0.7951**
VGG19-DNN-Cat	0.7702/**0.7931**	0.7714/**0.7969**	0.7681/**0.7936**	0.769/**0.7925**
VGG19-DNN-ElemMult	0.5505/**0.5873**	0.4183/**0.6473**	0.5391/**0.6046**	0.496/**0.5755**
VGG19-FC-Avg	**0.7335**/0.7110	**0.7328**/0.7110	**0.7328**/0.7100	**0.733**/0.7103
VGG19-FC-Cat	**0.7059**/0.7004	**0.7057**/0.7001	**0.7053**/0.7003	**0.705**/0.7000
VGG19-FC-ElemMult	0.5605/**0.5819**	0.5963/**0.6261**	0.5570/**0.5976**	0.526/**0.5726**
VGG19-IHAN(EA)-Avg	**0.7933**/0.7894	**0.7953**/0.7916	**0.7951**/0.7914	**0.793**/0.7893
VGG19-IHAN(EA)-Cat	**0.7944**/0.7904	**0.7963**/0.7926	**0.7954**/0.7930	**0.794**/0.7904
VGG19-IHAN(EA)-ElemMult	0.5914/**0.5998**	0.6594/**0.6734**	0.6004/**0.6184**	0.570/**0.5864**
VGG19-IHAN-Avg	0.7903/**0.7926**	0.7905/**0.7935**	0.7903/**0.7940**	0.790/**0.7924**
VGG19-IHAN-Cat	**0.8073**/0.7930	**0.8084**/0.7934	**0.8090**/0.7942	**0.807**/0.7928
VGG19-IHAN-ElemMult	0.6014/**0.6023**	0.6742/**0.6803**	0.6199/**0.6212**	0.588/**0.5884**
VGG19-MaxPool-Avg	**0.7807**/0.7795	**0.7808**/0.7794	**0.7813**/0.7799	**0.780**/0.7793
VGG19-MaxPool-Cat	**0.8034**/0.7987	**0.8041**/0.7995	**0.8044**/0.8002	**0.803**/0.7986
VGG19-MaxPool-ElemMult	**0.5984**/0.5977	0.6681/**0.6793**	0.6117/**0.6171**	0.581/**0.5827**

Table A.4: Performance comparison between FakeNewsNet+ and FakeNewsNet for all the I-KAHAN configurations on the PolitiFact dataset. The numbers on the left side of each slash are those of FakeNewsNet+, while those on the right-hand side are of the original FakeNewsNet. The highest numbers are in bold.

Configuration	**PolitiFact**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP(EA)-Cat	0.8186/**0.8346**	0.8231/**0.8365**	0.8125/**0.8303**	0.809/**0.8301**
CLIP-Cat	0.9020/**0.9097**	0.9059/**0.9121**	0.8983/**0.9067**	0.901/**0.9088**
Resnet50-AvgPool-Avg	0.8713/0.8713	0.8728/0.8728	0.8691/0.8691	0.870/**0.8702**
Resnet50-AvgPool-Cat	0.8875/0.8875	0.8873/0.8873	0.8863/0.8863	**0.887**/0.8867
Resnet50-AvgPool-ElemMult	0.6735/0.6735	0.6636/0.6636	0.6729/0.6729	0.659/0.6590
Resnet50-DNN-Avg	0.8611/**0.8654**	0.8674/**0.8721**	0.8541/**0.8586**	0.859/**0.8633**
Resnet50-DNN-Cat	0.8653/**0.8730**	0.8677/**0.8760**	0.8684/**0.8766**	0.865/**0.8726**
Resnet50-DNN-ElemMult	0.6498/0.6498	0.6913/0.6913	0.6546/0.6546	0.637/**0.6374**
Resnet50-FC-Avg	0.7852/0.7852	0.7844/0.7844	0.7846/0.7846	0.784/0.7840
Resnet50-FC-Cat	0.7894/0.7894	0.7897/0.7897	0.7921/0.7921	**0.789**/0.7887
Resnet50-FC-ElemMult	0.6303/0.6303	0.6815/0.6815	0.6471/0.6471	**0.618**/0.6175
Resnet50-IHAN(EA)-Avg	0.8628/0.8628	0.8626/0.8626	0.8607/0.8607	**0.862**/0.8617
Resnet50-IHAN(EA)-Cat	0.8876/0.8876	0.8911/0.8911	0.8849/0.8849	0.886/**0.8864**
Resnet50-IHAN(EA)-ElemMult	**0.6387**/0.6146	**0.5826**/0.5685	**0.6595**/0.6414	**0.613**/0.5880
Resnet50-IHAN-Avg	0.8713/0.8713	0.8709/0.8709	0.8685/0.8685	0.870/**0.8702**
Resnet50-IHAN-Cat	0.8850/**0.8858**	0.8859/**0.8866**	0.8836/**0.8845**	0.884/**0.8850**
Resnet50-IHAN-ElemMult	0.6827/0.6827	0.6141/0.6141	0.6781/0.6781	0.652/**0.6521**
Resnet50-MaxPool-Avg	0.8090/0.8090	0.8093/0.8093	0.8049/0.8049	0.807/**0.8073**
Resnet50-MaxPool-Cat	0.8790/0.8790	0.8796/0.8796	0.8770/0.8770	**0.878**/0.8779
Resnet50-MaxPool-ElemMult	0.7519/0.7519	0.7900/0.7900	0.7465/0.7465	**0.745**/0.7447
VGG19-AvgPool-Avg	**0.8756**/0.7971	**0.8749**/0.8100	**0.8736**/0.7992	**0.875**/0.7940
VGG19-AvgPool-Cat	**0.8918**/0.8595	**0.8922**/0.8597	**0.8888**/0.8543	**0.891**/0.8575
VGG19-AvgPool-ElemMult	**0.7016**/0.6650	**0.6977**/0.6649	**0.7028**/0.6624	**0.689**/0.6438
VGG19-DNN-Avg	**0.8645**/0.8372	**0.8733**/0.8580	**0.8557**/0.8248	**0.862**/0.8311
VGG19-DNN-Cat	**0.8816**/0.8364	**0.8854**/0.8426	**0.8772**/0.8272	**0.880**/0.8330
VGG19-DNN-ElemMult	**0.6751**/0.6385	**0.7219**/0.6735	**0.6522**/0.6070	**0.653**/0.5984
VGG19-FC-Avg	0.7895/0.7895	0.7891/0.7891	0.7859/0.7859	**0.788**/0.7876
VGG19-FC-Cat	0.7861/0.7861	0.7840/0.7840	0.7835/0.7835	0.784/**0.7844**
VGG19-FC-ElemMult	0.6147/0.6147	0.6734/0.6734	0.6369/0.6369	**0.603**/0.6025
VGG19-IHAN(EA)-Avg	0.8594/0.8594	0.8586/0.8586	0.8579/0.8579	0.858/**0.8585**
VGG19-IHAN(EA)-Cat	0.8799/0.8799	0.8798/0.8798	0.8793/0.8793	0.879/**0.8791**
VGG19-IHAN(EA)-ElemMult	0.6463/0.6463	0.5938/0.5938	0.6640/0.6640	0.620/0.6200
VGG19-IHAN-Avg	0.8730/0.8730	0.8729/0.8729	0.8701/0.8701	**0.872**/0.8718
VGG19-IHAN-Cat	0.8773/0.8773	0.8789/0.8789	0.8734/0.8734	0.876/0.8760
VGG19-IHAN-ElemMult	0.6844/0.6844	0.6162/0.6162	0.6807/0.6807	**0.655**/0.6546
VGG19-MaxPool-Avg	**0.8747**/0.8141	**0.8731**/0.8095	**0.8726**/0.8079	**0.874**/0.8101
VGG19-MaxPool-Cat	0.8926/0.8926	0.8924/0.8924	0.8911/0.8911	**0.892**/0.8918
VGG19-MaxPool-ElemMult	**0.7024**/0.6725	**0.7425**/0.6768	**0.7077**/0.6834	**0.696**/0.6601

Table A.5: Performance comparison between FakeNewsNet+ and FakeNewsNet for all the I-KAHAN configurations on the GossipCop dataset. The numbers on the left side of each slash are those of FakeNewsNet+, while those on the right-hand side are of the original FakeNewsNet. The highest numbers are in bold.

Configuration	**GossipCop**			
	Accuracy	**Precision**	**Recall**	**F1**
CLIP(EA)-Cat	0.7657/**0.7672**	0.7669/**0.7677**	0.7650/**0.7669**	0.765/**0.7667**
CLIP-Cat	0.8011/**0.8046**	0.8020/**0.8045**	0.8001/**0.8044**	0.800/**0.8042**
Resnet50-AvgPool-Avg	0.7932/0.7932	0.7934/0.7934	0.7935/0.7935	**0.793**/0.7929
Resnet50-AvgPool-Cat	**0.7978**/0.7903	**0.7973**/0.7903	**0.7980**/0.7896	**0.798**/0.7898
Resnet50-AvgPool-ElemMult	**0.6025**/0.5914	**0.6803**/0.6555	**0.6214**/0.6045	**0.589**/0.5743
Resnet50-DNN-Avg	**0.8242**/0.7751	**0.8247**/0.7759	**0.8222**/0.7736	**0.824**/0.7743
Resnet50-DNN-Cat	**0.8073**/0.7907	**0.8075**/0.7907	**0.8069**/0.7894	**0.807**/0.7901
Resnet50-DNN-ElemMult	0.5505/**0.5509**	0.4541/**0.4546**	0.5388/**0.5392**	0.496/**0.4966**
Resnet50-FC-Avg	**0.6867**/0.6855	**0.6858**/0.6846	**0.6841**/0.6828	**0.685**/0.6842
Resnet50-FC-Cat	**0.6757**/0.6749	**0.6767**/0.6759	**0.6740**/0.6732	**0.674**/0.6735
Resnet50-FC-ElemMult	**0.5671**/0.5664	**0.5650**/0.5350	**0.5676**/0.5663	**0.537**/0.5331
Resnet50-IHAN(EA)-Avg	0.7901/**0.7917**	0.7918/**0.7920**	0.7920/**0.7931**	0.790/**0.7916**
Resnet50-IHAN(EA)-Cat	0.7840/**0.7925**	0.7853/**0.7941**	0.7852/**0.7945**	0.784/**0.7924**
Resnet50-IHAN(EA)-ElemMult	0.5911/**0.5949**	0.6561/**0.6626**	0.5999/**0.6084**	0.570/**0.5773**
Resnet50-IHAN-Avg	**0.7950**/0.7847	**0.7955**/0.7856	**0.7964**/0.7859	**0.795**/0.7845
Resnet50-IHAN-Cat	**0.7959**/0.7957	**0.7965**/0.7963	**0.7968**/0.7966	**0.796**/0.7955
Resnet50-IHAN-ElemMult	**0.5962**/0.5898	**0.6639**/0.6503	**0.6099**/0.6024	**0.579**/0.5732
Resnet50-MaxPool-Avg	**0.7625**/0.7549	**0.7618**/0.7542	**0.7621**/0.7541	**0.762**/0.7544
Resnet50-MaxPool-Cat	**0.7872**/0.7797	**0.7877**/0.7812	**0.7877**/0.7808	**0.787**/0.7795
Resnet50-MaxPool-ElemMult	**0.5998**/0.5977	**0.6702**/0.6622	**0.6133**/0.6105	**0.582**/0.5810
VGG19-AvgPool-Avg	0.7920/0.7920	0.7923/0.7923	0.7925/0.7925	**0.792**/0.7917
VGG19-AvgPool-Cat	0.7977/0.7977	0.7978/0.7978	0.7984/0.7984	0.797/**0.7975**
VGG19-AvgPool-ElemMult	0.5944/0.5944	0.6627/0.6627	0.6079/0.6079	**0.577**/0.5766
VGG19-DNN-Avg	0.7667/**0.7668**	**0.7709**/0.7708	0.7634/**0.7635**	0.765/**0.7651**
VGG19-DNN-Cat	0.7702/0.7702	0.7714/0.7714	0.7681/0.7681	0.769/**0.7691**
VGG19-DNN-ElemMult	0.5505/0.5505	0.4183/0.4183	0.5391/0.5391	**0.496**/0.4956
VGG19-FC-Avg	**0.7335**/0.7134	**0.7328**/0.7126	**0.7328**/0.7125	**0.733**/0.7129
VGG19-FC-Cat	0.7059/0.7059	0.7057/0.7057	0.7053/0.7053	0.705/**0.7053**
VGG19-FC-ElemMult	**0.5605**/0.5597	**0.5963**/0.5928	**0.5570**/0.5557	**0.526**/0.5256
VGG19-IHAN(EA)-Avg	**0.7933**/0.7932	**0.7953**/0.7952	**0.7951**/0.7950	0.793/0.7930
VGG19-IHAN(EA)-Cat	**0.7944**/0.7927	**0.7963**/0.7941	**0.7954**/0.7940	**0.794**/0.7925
VGG19-IHAN(EA)-ElemMult	**0.5914**/0.5902	0.6594/**0.6625**	**0.6004**/0.5986	**0.570**/0.5654
VGG19-IHAN-Avg	**0.7903**/0.7885	**0.7905**/0.7888	**0.7903**/0.7891	**0.790**/0.7883
VGG19-IHAN-Cat	**0.8073**/0.7972	**0.8084**/0.7981	**0.8090**/0.7985	**0.807**/0.7970
VGG19-IHAN-ElemMult	**0.6014**/0.5939	**0.6742**/0.6292	**0.6199**/0.6068	**0.588**/0.5720
VGG19-MaxPool-Avg	0.7807/0.7807	0.7808/0.7808	0.7813/0.7813	0.780/**0.7805**
VGG19-MaxPool-Cat	**0.8034**/0.7928	**0.8041**/0.7935	**0.8044**/0.7932	**0.803**/0.7925
VGG19-MaxPool-ElemMult	0.5984/0.5984	0.6681/0.6681	0.6117/0.6117	**0.581**/0.5809

NTNU
Norwegian University of
Science and Technology