Understanding Disinformation: Learning with Weak Social Supervision

by

Kai Shu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2020 by the
Graduate Supervisory Committee:

Huan Liu, Chair
H. Russell Bernard
Ross Maciejewski
Guoliang Xue

ARIZONA STATE UNIVERSITY

August 2020

ABSTRACT

Social media has become an important means of user-centered information sharing and communications in a gamut of domains, including news consumption, entertainment, marketing, public relations, and many more. The low cost, easy access, and rapid dissemination of information on social media draws a large audience but also exacerbate the wide propagation of disinformation including fake news, i.e., news with intentionally false information. Disinformation on social media is growing fast in volume and can have detrimental societal effects. Despite the importance of this problem, our understanding of disinformation in social media is still limited. Recent advancements of computational approaches on detecting disinformation and fake news have shown some early promising results. Novel challenges are still abundant due to its complexity, diversity, dynamics, multi-modality, and costs of fact-checking or annotation.

Social media data opens the door to interdisciplinary research and allows one to collectively study large-scale human behaviors otherwise impossible. For example, user engagements over information such as news articles, including posting about, commenting on, or recommending the news on social media, contain abundant rich information. Since social media data is big, incomplete, noisy, unstructured, with abundant social relations, solely relying on user engagements can be sensitive to noisy user feedback. To alleviate the problem of limited labeled data, it is important to combine contents and this new (but weak) type of information as supervision signals, i.e., *weak social supervision*, to advance fake news detection.

The goal of this dissertation is to understand disinformation by proposing and exploiting weak social supervision for learning with little labeled data and effectively detect disinformation via innovative research and novel computational methods. In

particular, I investigate learning with weak social supervision for understanding disinformation with the following computational tasks: bringing the heterogeneous social context as auxiliary information for *effective fake news detection*; discovering explanations of fake news from social media for *explainable fake news detection*; modeling multi-source of weak social supervision for *early fake news detection*; and transferring knowledge across domains with adversarial machine learning for *cross-domain fake news detection*. The findings of the dissertation significantly expand the boundaries of disinformation research and establish a novel paradigm of learning with weak social supervision that has important implications in broad applications in social media.

DEDICATION

I dedicate my dissertation work to my loving parents, Yuanshan Shu and Qiuai Hu, for supporting me to pursue my dream!

I also dedicate this dissertation to my wife Ling Luo for supporting me all the way!

Without their help, encouragement and accompany, this journey would have not been possible.

ACKNOWLEDGMENTS

This dissertation is impossible without the help from my advisor Dr. Huan Liu. I would like to thank him for giving me large freedom through my Ph.D. study to explore various research problems and his excellent advising skills with great patience and guidance, which makes my Ph.D. experience colorful, exciting and productive. I'm very fortunate to have Dr. Liu as my Ph.D. advisor, from whom I learned many abilities that can benefit all my life: how to write papers and give presentations, how to discover and address challenging and important problems, and how to establish my career and see the big vision. Dr. Liu is not only a supervisor for research but also a life mentor to help me improve my personality, overcome difficulties and make better decisions. He is extremely kind and helpful to give his suggestions and provide help for many aspects in my life and he is so generous to share his experience and knowledge to help me avoid detours in my life. Dr. Liu, I cannot thank you enough!

I would like to thank my committee members, Dr. H. Russell Bernard Dr. Ross Maciejewski, and Dr. Guoliang Xue, for helpful suggestions and insightful comments. I am very grateful to the committee members for their challenging questions during my comprehensive exam, which allow me think broad and deep for my current research and future research agenda. I always consider Dr. H. Russell Bernard as my secondary advisor because his broad interests and knowledge across disciplines largely broaden my perspectives on research and greatly expand the boundaries of my research. I took a graduate course on data visualization from Dr. Ross Maciejewski. His insightful discussions and comments provided me new angles to rethink beyond my research on weak supervision learning for disinformation. I also took the game theory course from Dr. Guoliang Xue, which prepared me with solid technical background and benefited my Ph.D. research a lot.

I was lucky to work as interns in Microsoft Research and Yahoo! Research with amazing colleagues and mentors: Susan Dumais, Ahmed Hassan Awadallah, Subhabrata Mukherjee, Guoqing Zheng, Milad Shokouhi, Wei Wang in Microsoft Research; Yunhong Zhou, Liangda Li, Yunzhong Liu, Yang Sun, Ruirui Li from Yahoo! Research. Because of you, my life became much easier in new environments; because of you, I enjoyed two wonderful and productive summers; and because of you, I was able to contribute my knowledge to exciting projects. Thank you for everything.

During my Ph.D. study, my friends and colleagues provided me consistent support and encouragement and they deserve a special thank. I am thankful to my colleagues at the Data Mining and Machine Learning Lab: Suhang Wang, Fred Morstatter, Deepak Mahudeswaran, Issac Jones, Justin Sampson, Tahora H. Nazer, Suhas Ranganath, Vineeth Rakesh, Jundong Li, Liang Wu, Ghazaleh Beigi, Kaize Ding, Lu Cheng, Ruocheng Guo, Yichuan Li, Nur Shazwani Kamrudin, Raha Moraffah, Matthew Davis, Ahmadreza Mosallanezhad, Amrita Bhattacharjee, Faisal Alatawi, Mansooreh Karami, Kumarage Tharindu, Paras Sheth, and Bohan Jiang. In particular, thanks to Suhang Wang who taught me how to do research; thanks to Fred Morstatter, Justin Sampson, and Isaac Jones as my English teachers and I will remember any error you corrected and every new word you taught; thanks to Reza Zafarani and Suhang Wang from whom I learned my presentation skills. I am also lucky to have you as friends and colleagues in my life: Ping Luo, Amy Sliva, Dongwon Lee, Juan Cao, Jiliang Tang, Xia Hu, Huiji Gao, Reza Zafarani, Nitin Agarwal, Xinyi Zhou, Yilin Wang, Xuying Meng, Shuo Yang, Wen Zhang, Qun Zhao, Qianru Wang, and Peifeng Yin.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Social media has become an important means of large-scale information sharing and communication in all occupations, including marketing, journalism, public relations, and more (Zafarani *et al.*, 2014). This change in consumption behaviors is due to some novel features such as mobility, free, and interactiveness. However, the low cost, easy access, and rapid dissemination of information of social media draw a large audience and enable the wide propagation of disinformation and *fake news*, i.e., news with intentionally false information.

The wide spread of disinformation and fake news can cause detrimental societal effects. First, people may accept deliberate lies as truths; second, fake news can change the way people respond to legitimate news; and third, the prevalence of fake news has the potential to break the trustworthiness of online journalism, and may lead to detrimental real-world consequences. For instance, in 2016, millions of people read and "liked" fake news stories proclaiming that Pope Francis has endorsed Donald Trump for U.S. president[1]. As a recent example, there are many reported cases for COVID-19 related fake news, such as the virus being a hoax, or 5G is the cause of the COVID-19[2]. Therefore, it becomes increasingly important for policy makers to regulate and discourage the creation of fake news, for online business to detect and prevent fake news, and for citizens to protect themselves from fake news.

[1]https://www.cnbc.com/2016/12/30/read-all-about-it-the-biggest-fake-news-stories-of-2016.html

[2]https://www.bbc.com/news/amp/stories-52731624

Despite the importance of the problem, our understanding of fake news is still limited. For example, we want to know why people create fake news, who produces and publishes them, how fake news spreads, what characteristics distinguish fake news from legitimate ones, or why some people are more susceptible to fake news than others (Mercier, 2017). Therefore, we propose to understand fake news with disciplines such as journalism, psychology, social science, and characterize the unique characteristics for its detection. Better understanding of fake news will allow us to come up with algorithmic solutions for *detecting* fake news and managing it before fake news is widely disseminated.

However, detecting disinformation and fake news poses unique challenges that makes it non-trivial. First, the *data challenge* has been a major roadblock because the content of fake news and disinformation is rather diverse in terms of topics, styles and media platforms; and fake news attempts to distort truth with diverse linguistic styles while simultaneously mocking true news. Thus, obtaining annotated fake news data is non-scalable, and data-specific embedding methods are not sufficient for fake news detection with little labeled data. Second, the *evolving* nature of fake news and disinformation is another obstacle in this task–fake news is usually related to newly emerging, time-critical events, which may not have been properly verified by existing knowledge bases (KB) due to the lack of corroborating evidence or claims.

Recently, learning with weak supervision has been of great interest the research community to mitigate the data scarcity problem for various tasks. Social media data allows one to study large-scale human behaviors, that make it suitable for deriving supervision signals. First, social media data is *big*. We have limited data for each individual. However, the social property of social media data links individuals' data together, which provides a new type of big data supervision. Second, social media

data is *linked*. The availability of social relations determines that social media data is inherently linked, meaning it is not independent and identically distributed. For example, user engagements over information such as news articles, including posting about, commenting on or recommending the news on social media, bear implicit judgments of the users to the news and could serve as auxiliary information for disinformation and fake news detection. In addition, social media data is *noisy* that make it a type of *weak* supervision. Users in social media can be both passive content consumers and active content producers, causing the quality of user-generated content to vary. Social networks are also noisy with the existence of malicious users such as spammers and bots.

This new (but weak) type of data mandates new computational analysis approaches that combine social theories and statistical data mining techniques. Due to the nature of social media engagements, we term these signals as *weak social supervision*.

Therefore, in this dissertation, I investigating the attempts of learning with weak social supervision to understand and detect disinformation and fake news on social media. In particular, I study the practical and challenging scenarios to detect disinformation and fake news more effectively, with explainability, at an early stage, and across domains. I propose novel frameworks to tackle these challenges and learn representations not only from textual content but also the unique properties of related social media engagements to detect fake news.

1.1 Research Challenges

To detect fake news on social media, we are faced with several challenges:

- For effective fake news detection, the social context during news dissemination

process on social media forms the inherent tri-relationship, the relationship among publishers, news pieces, and end users, which has potential to improve fake news detection. How can we mathematically model the tri-relationship to extract feature representations of news pieces? And how do we take advantage of tri-relationship modeling for fake news detection?

- For explainable fake news detection, both news sentences and user comments contain important cues to detect and explain fake news. First, news contents may contain information that is verifiably false. Second, user comments have rich information from the crowd on social media that are useful to detect fake news. Moreover, news contents and user comments inherently are related and can provide signals to explain why a given news article is fake or not. How can we perform explainable fake news detection that can achieve explainability and good detection performance simultaneously? And how can we model the correlation between news contents and user comments jointly for explainable fake news detection?
- For early fake news detection, multiple weak signals from different sources of social engagements contains complementary utilities in addition to news content to detect fake news early. For example, user engagements over news articles such as comments, bear implicit judgments of the users about the news and could serve as weak sources of labels. How can we leverage a limited amount of clean data along with weak signals from social engagements to train a fake news detector? And how can we model the weight of weak labels from different sources that regulate the learning process of the fake news classifier?
- For cross-domain fake news detection, it is important to explore auxiliary information to improve fake news prediction with limited labels for the newly

emerging target domain. Fake news publishers often have intent to spread distorted and misleading information widely, requiring particular writing styles that are domain-independent. In addition, rich user engagements within a single domain have significant complementary information in addition to news contents. How can we learn domain-adaptive news representations across news domains? And how can we capture news contents, user comments, and user-news interactions to detect fake news?

1.2 Contributions

The contributions of this dissertation are summarized as follows:

- Studying novel problems of understanding disinformation such as fake news on social media;
- Providing principled approaches to learn with weak social supervision guided by social theories for multi-faceted social media data;
- Proposing novel frameworks to detect fake news with challenging scenarios including effective, explainable, early, and cross-domain fake news detection; and
- Conducting experiments on real-world datasets to demonstrate the effectiveness of the proposed frameworks.

1.3 Organization

The remainder of this dissertation is organized as follows. In Chapter 2, I review related works in disinformation and fake news detection and learning with weak

social supervision. In Chapter 3, I investigate effective fake news detection. I first give the problem statement and details of the proposed framework tri-relationship embedding (TriFN), whose design is guided by sociological and journalism studies on publisher bias and user credibility. I then conduct experiments to evaluate the effectiveness of TriFN for fake news detection. In Chapter 4, I study explainable fake news detection. I first detail the proposed framework dEFEND by demonstrating how to use sentence-comment co-attention networks to discover explainable signals and effective features. Then I show extensive experimental results on the effectiveness of dEFEND. In Chapter 5, I study the early fake news detection. I first introduce how to model multi-source of weak labels with a small set of clean labels using MWSS. I then demonstrate the constructions of weak labels from user engagements in social media guided by social theories. Finally, I conduct extensive experiments to demonstrate the effectiveness of MWSS for fake news detection early. In Chapter 6, I propose CrossFND for cross-domain fake news detection. I first introduce how to jointly capture cross-domain knowledge transfer and with-in domain auxiliary information from with CrossFND for news representation learning. I then conduct experiments to evaluate CrossFND across different domains. I conclude the dissertation and point out broader impacts and promising research directions in Chapter 7.

Chapter 2

FOUNDATIONS AND PRELIMINARIES

In this chapter, I will briefly introduce the background about researchers in disinformation and fake news detection and learning with weak supervision.

2.1 Disinformation and Fake News Detection

Disinformation and misinformation have been an important issue and attracts increasing attention in recent years (Kumar *et al.*, 2016). The openness and anonymity of social media makes it convenient for users to share and exchange information, but also makes it vulnerable to nefarious activities. Though the spread of misinformation and disinformation has been studied in journalism, the openness of social networking platforms, combined with the potential for automation, facilitates the dis/misinformation to rapidly propagate to massive numbers of people, which brings about unprecedented challenges. Specifically, disinformation is fake or inaccurate information that is intentionally spread to mislead and/or deceive; misinformation is false content shared by a person who does not realize it is false or misleading. In addition, there are some other related types of information disorder (Wu *et al.*, 2019; Zhou and Zafarani, 2018): *rumor* is a story circulating from person to person, of which the truth is unverified or doubtful. Rumors usually arise in the presence of ambiguous or threatening events. When its statement is proved to be false, a rumor is a type of misinformation; *Urban Legend* is a fictional story that contains themes related to local popular culture. The statement and story of an urban legend are

usually false. An urban legend is usually describing unusual, humorous, or horrible events; *Spam* is unsolicited messages sent to a large number of recipients, containing irrelevant or inappropriate information, which is unwanted.

As a representative example of disinformation, we briefly introduce the related work about fake news detection on social media. Fake news detection methods generally focus on using *news contents* and *social contexts* (Shu *et al.*, 2017; Zhou *et al.*, 2019a).

News contents contain the clues to differentiate fake and real news. For news content based approaches, features are extracted as linguistic-based and visual-based. Linguistic-based features capture specific writing styles and sensational headlines that commonly occur in fake news content (Potthast *et al.*, 2017), such as lexical and syntactic features. Visual-based features try to identify fake images (Gupta *et al.*, 2013) that are intentionally created or capturing specific characteristics for images in fake news. News content based models include i) knowledge-based: using external sources to fact-checking claims in news content (Magdy and Wanas, 2010; Wu *et al.*, 2014), and 2) style-based: capturing the manipulators in writing style, such as deception (Feng *et al.*, 2012; Rubin and Lukoianova, 2015) and non-objectivity (Potthast *et al.*, 2017). For example, Potthast *et al.* (Potthast *et al.*, 2017) extracted various style features from news contents and predict fake news and media bias.

In addition to news content, social context related to news pieces contains rich information to help detect fake news. For social context based approaches, the features mainly include user-based, post-based and network-based. User-based features are extracted from user profiles to measure their characteristics and credibilities (Castillo *et al.*, 2011; Kwon *et al.*, 2013; Shu *et al.*, 2018c; Guacho *et al.*, 2018). For example, Shu *et al.* (Shu *et al.*, 2018c) proposed to understand user profiles from various aspects to differentiate fake news. Yang *et al.* (Yang *et al.*, 2019) proposed an unsupervised fake

news detection algorithm by utilizing users' opinions on social media and estimating their credibilities.

Post-based features represent users' social response in term of stance (Jin *et al.*, 2016), topics (Ma *et al.*, 2015), or credibility (Castillo *et al.*, 2011; Wu and Liu, 2018). Network-based features are extracted by constructing specific networks, such as diffusion network (Kwon *et al.*, 2013) etc. Social context models basically include stance-based and propagation-based. Stance-based models utilize users' opinions towards the news to infer news veracity (Jin *et al.*, 2016). Propagation-based models assume that the credibility of news is highly related to the credibilities of relevant social media posts, which several propagation methods can be applied (Jin *et al.*, 2016). Recently, deep learning models are applied to learn the temporal and linguistic representation of news (Shu *et al.*, 2019b; Wang *et al.*, 2018; Karimi *et al.*, 2018). Recently, research also focuses on challenging problems of fake news detection, such as fake news early detection by adversarial learning (Wang *et al.*, 2018) and user response generating (Qian *et al.*, 2018).

Despite the success of aforementioned fake news detection algorithms, they mostly rely on large amounts of labeled instances to train supervised models. Such large labeled training data is often difficult to obtain for disinformation and fake news. Therefore, I study novel algorithms to learn with weak social supervision for detecting fake news effectively, with explanation, at an early stage, and across domains.

2.2 From Weak Supervision to Weak Social Supervision

Learning with weak supervision is an important and newly emerging research area, and there are different ways of defining and approaching the problem. One definition

of weak supervision is leveraging higher-level and/or noisier input from subject matter experts (SMEs). The supervision from SMEs are represented in the form of weak label distributions, which mainly come from the following sources: 1) *inexact supervision*: a higher-level and coarse-grained supervision; 2) *inaccurate supervision*: a low-quality and noisy supervision; and 3) *existing resources*: using existing resources to provide supervision. Another definition categorizes weak supervision into inexact supervision, inaccurate supervision, and incomplete supervision (Zhou, 2018). The incomplete supervision means that a subset of training data are given with labels, which essentially includes active learning and semi-supervised learning techniques. Weak supervision can be formed in deterministic (e.g., in the form of *weak labels*) and non-deterministic (e.g., in the form of *constraints*) ways.

Incorporating Weak Labels: Most machine learning models rely on the scale of labeled data to achieve good performance where the presence of label noise (Nettleton *et al.*, 2010) or adversarial noise (Reed *et al.*, 2014) can cause a dramatic performance drop. Therefore, learning with noisy labels has been of great interest to the research community for various tasks (Shu *et al.*, 2020b; Frénay and Verleysen, 2013; Meng *et al.*, 2019). Some of the existing works attempt to rectify the weak labels by incorporating a loss correction mechanism (Sukhbaatar *et al.*, 2014; Patrini *et al.*, 2017). Sukhbaatar *et al.* (Sukhbaatar *et al.*, 2014) introduce a linear layer to adjust the loss and estimate label corruption with access to the true labels (Sukhbaatar *et al.*, 2014). Patrini *et al.* (Patrini *et al.*, 2017) utilize the loss correction mechanism to estimate a label corruption matrix without making use of clean labels. Other works consider the scenario where a small set of clean labels are available (Li *et al.*, 2017; Zheng *et al.*, 2019; Hendrycks *et al.*, 2018; Ren *et al.*, 2018). For example, Veit *et al.* use human-verified labels and train a label cleaning network in a multi-label classification setting;

Zheng *et al.* propose a meta label correction approach using a meta model which provides reliable labels for the main models to learn. Recent works also consider the scenario where weak signals are available from multiple sources (Ratner *et al.*, 2017; Varma *et al.*, 2019; Ratner *et al.*, 2018a) to exploit the redundancy as well as the consistency in the labeling information.

Injecting Constraints: Directly learning with weak labels may suffer from the noisy label problem. Instead, representing weak supervision as constraints can avoid noisy labels and encode domain knowledge into the learning process of prediction function. The constraints can be injected over the output space and/or the input representation space. For example, Stewart *et al.* (Stewart and Ermon, 2017) model prior physics knowledge on the outputs to penalize "structures" that are not consistent with the prior knowledge. For relation extraction tasks, label-free distant supervision can be achieved via encoding entity representations under transition law from knowledge bases (KB). This type of weak supervision, i.e., injecting constraints, is often based on prior knowledge from domain experts, which are *jointly* optimized with the primary learning objective of prediction tasks.

With the rise of social media, the web has become a vibrant and lively realm where billions of individuals all around the globe interact, share, post and conduct numerous daily activities. Social media enables us to be connected and interact with anyone, anywhere and anytime, which allows us to observe human behaviors in an unprecedented scale with a new lens. However, significantly different from traditional data, social media data is big, incomplete, noisy, unstructured, with abundant social relations. This new type of data contains rich *social interactions* that can provide additional signals for obtaining weak supervision. Generally, there are three major aspects of the social media engagements: users, contents, and relations (Shu

et al., 2020a,c). First, users exhibit different characteristics that indicate different patterns of behaviors. Second, users express their opinions and emotions through posts/comments. Third, users form different types of relations on social media through various communities. Therefore, social media data provides a new type of weak supervision, i.e., *weak social supervision*, which has great potentials to advance a wide range of applications including fake news detection. We build on top of previous work in the area of learning from weak supervision, and propose a new paradigm called weak social supervision.

Chapter 3

EFFECTIVE FAKE NEWS DETECTION

In this chapter, we investigate effective fake news detection with social context. The news ecosystem on social media provides abundant social context information, which involves three basic entities, i.e., publishers, news pieces, and social media users. Figure 1 gives an illustration of such ecosystem. In Figure 1, p_1, p_2 and p_3 are news publishers who publish news $a_1, \ldots, a_4$ and $u_1, \ldots, u_6$ are users who have engaged in sharing these news pieces. In addition, users tend to form social links with like-minded people with similar interests. The *tri-relationship*, the relationship among publishers, news pieces, and users, contains additional information to help detect fake news.

First, sociological studies on journalism have theorized the correlation between the partisan bias of publisher and the veracity degree of news content (Gentzkow *et al.*, 2014). For example, in Figure 1, p_1 is a publisher with extreme left partisan bias and p_2 is a publisher with extreme right partisan bias. To support their own partisan viewpoints, they have high degree to distort the facts and report fake news pieces, such as a_1 and a_3; while for a mainstream publisher p_3 that has least partisan bias, he/she has a lower chance to manipulate original news events, and is more likely to write a true news piece a_4.

Second, mining user engagements towards news pieces on social media also help fake news detection. However, on social media, different users have different credibility levels. Those less credible users, such as malicious accounts or normal users who are vulnerable to fake news, are more likely to spread fake news. For example, u_2 and u_4 are users with low credibility scores, and they tend to spread fake news more

than other highly credible users. In addition, users tend to form relationships with like-minded people (Quattrociocchi *et al.*, 2016). For example, user u_5 and u_6 are friends on social media, so they tend to post those news that confirm their own views, such as a_4.

Moreover, the publisher-news relationships and user-news interactions both provide new and different perspectives of social context, and thus contain complementary information to advance fake news detection. In an attempt to model the tri-relationship, we face the following challenges: (1) how to mathematically model the tri-relationship to extract feature representations of news pieces; and (2) how to take advantage of tri-relationship modeling for fake news detection. To solve these two challenges, I proposed a novel framework Tri-relationship for Fake News detection (TriFN), which will be introduced in detail next.

3.1 Problem Statement

Let $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ be the set of n news pieces, and $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ be the set of m users on social media posting these news pieces. We denote $\mathbf{X} \in \mathbb{R}^{n \times t}$ as the bag-of-word feature matrix of news pieces, where t is the dimension of vocabulary size. We use $\mathbf{A} \in \{0, 1\}^{m \times m}$ to denote the user-user adjacency matrix, where $\mathbf{A}_{ij} = 1$ indicates that user u_i and u_j are friends; otherwise $\mathbf{A}_{ij} = 0$. We denote the user-news interaction matrix as $\mathbf{W} \in \{0, 1\}^{m \times n}$, where $\mathbf{W}_{ij} = 1$ indicates that user u_i has shared the news piece a_j; otherwise $\mathbf{W}_{ij} = 0$. It's worth mentioning that we focus on those user-news interactions in which users agree with the news. For example, we only consider those users who share news pieces without comments, and these users share the same alignment of viewpoints with the news items. We will introduce more details

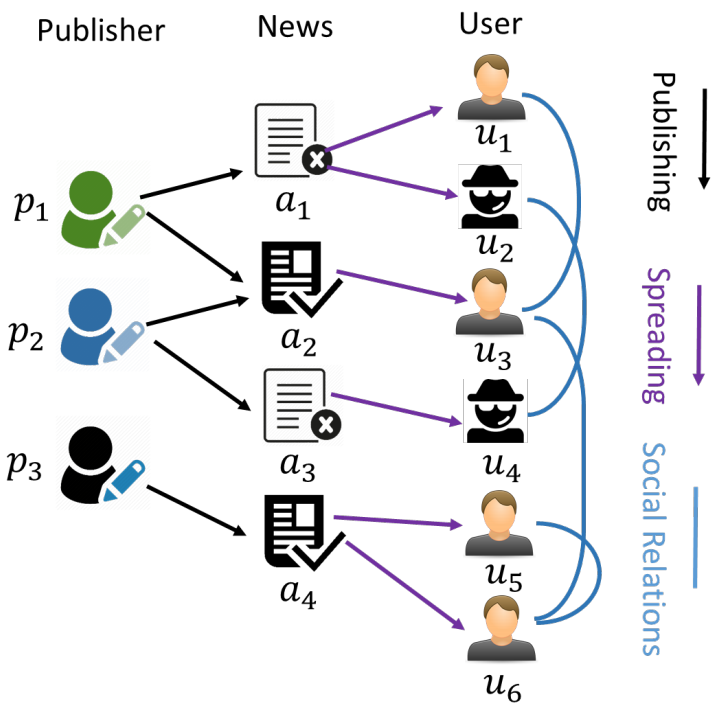


Figure 1: An illustration of tri-relationship among publishers, news pieces, and users, during the news dissemination process.

in Section 3.2.3. We also denote $\mathcal{P} = \{p_1, p_2, ..., p_l\}$ as the set of l news publishers. In addition, we denote $\mathbf{B} \in \mathbb{R}^{l \times n}$ as the publisher-news publishing matrix, and $\mathbf{B}_{kj} = 1$ means news publisher p_k publishes the news article a_j; otherwise $\mathbf{B}_{kj} = 0$. We assume that the partisan bias labels of some publishers are given and available (see more details of how to collect partisan bias labels in Sec 3.2.4). We define $\mathbf{o} \in \{-1, 0, 1\}^{l \times 1}$ as the partisan label vectors, where -1, 0, 1 represents left-, neutral-, and right-partisan bias.

Similar to previous research (Shu *et al.*, 2017; Jin *et al.*, 2016), we treat fake news detection problem as a binary classification problem. In other words, each news piece can be true or fake, and we use $\mathbf{y} = \{\mathbf{y}_1; \mathbf{y}_2; ...; \mathbf{y}_n\} \in \mathbb{R}^{n \times 1}$ to represent the labels, and $\mathbf{y}_j = 1$ means news piece a_j is fake news; $\mathbf{y}_j = -1$ means true news. With the notations given above, the problem is formally defined as follows.

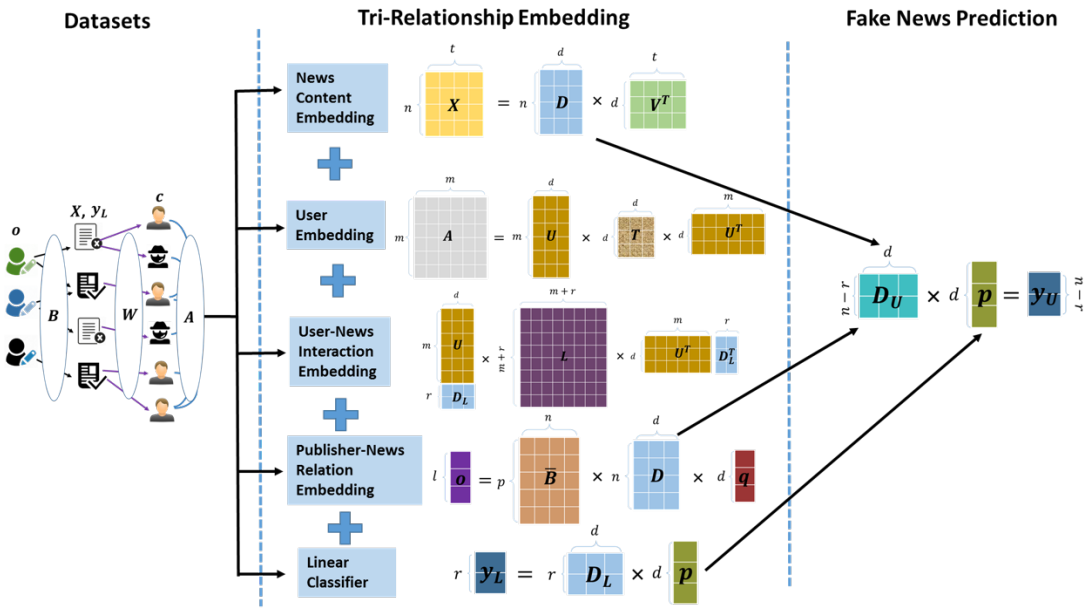


Figure 2: The tri-relationship embedding framework, which consists of five components: news contents embedding, user embedding, user-news interaction embedding, publisher-news relation embedding, and news classification.

Problem Statement: Given news article feature matrix $\mathbf{X}$, user adjacency matrix $\mathbf{A}$, user social engagement matrix $\mathbf{W}$, publisher-news publishing matrix $\mathbf{B}$, publisher partisan label vector $\mathbf{o}$, and partial labeled news vector $\mathbf{y}_L$, we aim to predict remaining unlabeled news label vector $\mathbf{y}_U$.

3.2 The Proposed Framework TriFN

In this section, we present the details of the proposed framework TriFN for modeling tri-relationship for fake news detection. It consists of five major components (Figure 2): a news contents embedding component, a user embedding component, a user-news interaction embedding component, a publisher-news relation embedding component, and a semi-supervised classification component.

In general, the news contents embedding component describes the mapping of news from bag-of-word features to latent feature space; the user embedding component illustrates the extraction of user latent features from user social relations; the user-news interaction embedding component learn the feature representations of news pieces guided by their partial labels and user credibilities; The publisher-news relation embedding component regularize the feature representations of news pieces through publisher partisan bias labels; The semi-supervised classification component learns a classification function to predict unlabeled news items.

3.2.1 News Contents Embedding

We can use news contents to find clues to differentiate fake news and true news. Recently, it has been shown that nonnegative matrix factorization (NMF) algorithms are very practical and popular to learn document representations (Xu *et al.*, 2003; Shahnaz *et al.*, 2006; Pauca *et al.*, 2004). It can project the news-word matrix $\mathbf{X}$ to a joint latent semantic factor space with low dimensionality, such that the news-word relations are modeled as the inner product in the space. Specifically, giving the news-word matrix $\mathbf{X} \in \mathbb{R}^{n \times t}$, NMF methods try to find two nonnegative matrices $\mathbf{D} \in \mathbb{R}_+^{n \times d}$ and $\mathbf{V} \in \mathbb{R}_+^{t \times d}$, where d is the dimension of the latent space, by solving the following optimization problem,

$$\min_{\mathbf{D},\mathbf{V} \geq 0} \| \mathbf{X} - \mathbf{D}\mathbf{V}^T \|_F^2 + \lambda(\|\mathbf{D}\|_F^2 + \|\mathbf{V}\|_F^2) \tag{3.1}$$

where $\mathbf{D}$ and $\mathbf{V}$ are the nonnegative matrices indicating low dimension representations of news pieces and words. Note that we denote $\mathbf{D} = [\mathbf{D}_L; \mathbf{D}_U]$, where $\mathbf{D}_L \in \mathbb{R}^{r \times d}$ is the news latent feature matrix for labeled news; while $\mathbf{D}_U \in \mathbb{R}^{(n-r) \times d}$ is the news

latent feature matrix for unlabeled news. The term $\lambda(\|\mathbf{D}\|_F^2 + \|\mathbf{V}\|_F^2)$ is introduced to avoid over-fitting.

3.2.2 User Embedding

On social media, people tend to form relationships with like-minded people, rather than those users who have opposing preferences and interests. Thus, connected users are more likely to share similar latent interests in news pieces. To obtain a standardized representation, we use nonnegative matrix factorization to learn the users' latent representations. Specifically, giving user-user adjacency matrix $\mathbf{A} \in \{0,1\}^{m \times m}$, we learn nonnegative matrix $\mathbf{U} \in \mathbb{R}_+^{m \times d}$ by solving the following optimization problem,

$$\min_{\mathbf{U},\mathbf{T} \geq 0} \|\mathbf{Y} \odot (\mathbf{A} - \mathbf{U}\mathbf{T}\mathbf{U}^T)\|_F^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{T}\|_F^2) \tag{3.2}$$

where $\mathbf{U}$ is the user latent matrix, $\mathbf{T} \in \mathbb{R}_+^{d \times d}$ is the user-user correlation matrix, $\mathbf{Y} \in \mathbb{R}^{m \times m}$ controls the contribution of $\mathbf{A}$, and $\odot$ denotes the Hadamard product operation. Since only positive links are observed in $\mathbf{A}$, following common strategies (Pan and Scholz, 2009), we first set $\mathbf{Y}_{ij} = 1$ if $\mathbf{A}_{ij} = 1$, and then perform negative sampling and generate the same number of unobserved links and set weights as 0. The term $\lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{T}\|_F^2)$ is to avoid over-fitting.

3.2.3 User-News Interaction Embedding

We model the user-news interactions by considering the relationships between user features and the labels of news items. We have shown that users with low credibilities are more likely to spread fake news, while users with high credibilities are less likely to spread fake news. To measure user credibility scores, we adopt the practical approach

in (Abbasi and Liu, 2013). The basic idea in (Abbasi and Liu, 2013) is that less credible users are more likely to coordinate with each other and form big clusters, while more credible users are likely to from small clusters. Specifically, the credibility scores are measured through the following major steps: 1) detect and cluster coordinate users based on user similarities; 2) weight each cluster based on the cluster size. Note that for our fake news detection task, we do not assume that credibility scores are directly provided, but inferred from widely available data, such as user-generated contents. By using the method in (Abbasi and Liu, 2013), we can assign each user u_i a credibility score $\mathbf{c}_i \in [0, 1]$. A larger $\mathbf{c}_i$ indicates that user u_i has a higher credibility, while a lower $\mathbf{c}_i$ indicates a lower credibility score. We use $\mathbf{c} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m\}$ to denote the credibility score vector for all users.

First, high-credibility users are more likely to share true news pieces, so we ensure that the distance between latent features of high-credibility users and that of true news is minimized,

$$\min_{\mathbf{U},\mathbf{D_L}\geq 0} \sum_{i=1}^{m}\sum_{j=1}^{r} \mathbf{W}_{ij}\mathbf{c}_i(1 - \frac{1+\mathbf{y}_{Lj}}{2})||\mathbf{U}_i - \mathbf{D}_{L_j}||_2^2 \tag{3.3}$$

and $(1 - \frac{1+\mathbf{y}_{Lj}}{2})$ is to ensure we only include true news pieces (i.e., $\mathbf{y}_{Lj} = -1$), and $\mathbf{c}_i$ is to adjust the contribution of user u_i to the loss function. For example, if $\mathbf{c}_i$ is large (high-credibility) and $\mathbf{W}_{ij} = 1$, we put a bigger weight on forcing the distance of feature $\mathbf{U}_i$ and $\mathbf{D}_{Lj}$ to be small; if $\mathbf{c}_i$ is small (low-credibility) and $\mathbf{W}_{ij} = 1$, than we put a smaller weight on forcing the distance of feature $\mathbf{U}_i$ and $\mathbf{D}_{Lj}$ to be small.

Second, low-credibility users are more likely to share fake news pieces, and we aim to minimize the distance between latent features of low-credibility users and that of fake news,

$$\min_{\mathbf{U},\mathbf{D_L}\geq 0} \sum_{i=1}^{m}\sum_{j=1}^{r} \mathbf{W}_{ij}(1-\mathbf{c}_i)(\frac{1+\mathbf{y}_{Lj}}{2})||\mathbf{U}_i - \mathbf{D}_{L_j}||_2^2 \tag{3.4}$$

and the term $(\frac{1+\mathbf{y}_{Lj}}{2})$ is to ensure we only include fake news pieces (i.e., $\mathbf{y}_{L_j} = 1$), and $(1-\mathbf{c}_i)$ is to adjust the contribution of user u_i to the loss function. For example, if $\mathbf{c}_i$ is large (high-credibility) and $\mathbf{W}_{ij} = 1$, we put a smaller weight on forcing the distance of feature $\mathbf{U}_i$ and $\mathbf{D}_{Lj}$ to be small; if $\mathbf{c}_i$ is small (low-credibility) and $\mathbf{W}_{ij} = 1$, then we put a bigger weight on forcing the distance of feature $\mathbf{U}_i$ and $\mathbf{D}_{Lj}$ to be small.

Finally, We combine Eqn 3.3 and Eqn 3.4 to consider the above two situations, and obtain the following objective function,

$$\begin{aligned}\min_{\mathbf{U},\mathbf{D_L}\geq 0} & \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{r} \mathbf{W}_{ij}\mathbf{c}_i(1-\frac{1+\mathbf{y}_{Lj}}{2})||\mathbf{U}_i - \mathbf{D}_{L_j}||_2^2}_{\text{True news}} \\ & + \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{r} \mathbf{W}_{ij}(1-\mathbf{c}_i)(\frac{1+\mathbf{y}_{Lj}}{2})||\mathbf{U}_i - \mathbf{D}_{L_j}||_2^2}_{\text{Fake news}}\end{aligned} \tag{3.5}$$

For simplicity, Eqn 3.5 can be rewritten as,

$$\min_{\mathbf{U},\mathbf{D_L}\geq 0} \sum_{i=1}^{m}\sum_{j=1}^{r} \mathbf{G}_{ij}||\mathbf{U}_i - \mathbf{D}_{L_j}||_2^2 \tag{3.6}$$

where $\mathbf{G}_{ij} = \mathbf{W}_{ij}(\mathbf{c}_i(1-\frac{1+\mathbf{y}_{Lj}}{2}) + (1-\mathbf{c}_i)(\frac{1+\mathbf{y}_{Lj}}{2}))$. If we denote a new matrix $\mathbf{H} = [\mathbf{U}; \mathbf{D}_L] \in \mathbb{R}^{(m+r)\times d}$, we can also rewrite Eqn. 3.6 as a matrix form as follows,

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{D_L}\geq 0} \sum_{i=1}^{m}\sum_{j=1}^{r} \mathbf{G}_{ij}||\mathbf{U}_i - \mathbf{D}_{L_j}||_2^2 \Leftrightarrow \min_{\mathbf{H}\geq 0} \sum_{i=1}^{m}\sum_{j=1+m}^{r+m} \mathbf{G}_{ij}||\mathbf{H}_i - \mathbf{H}_j||_2^2 \\ &\Leftrightarrow \min_{\mathbf{H}\geq 0} \sum_{i,j=1}^{m+r} \mathbf{F}_{ij}||\mathbf{H}_i - \mathbf{H}_j||_2^2 \Leftrightarrow \min_{\mathbf{H}\geq 0} \text{tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})\end{aligned} \tag{3.7}$$

where $\mathbf{L} = \mathbf{S} - \mathbf{F}$ is the Laplacian matrix and $\mathbf{S}$ is a diagonal matrix with diagonal

element $S_{ii} = \sum_{j=1}^{m+r} \mathbf{F}_{ij}$. $\mathbf{F} \in \mathbb{R}^{(m+r)\times(m+r)}$ is computed as follows,

$$\mathbf{F}_{ij} = \begin{cases} 0, & i,j \in [1,m] \text{ or } i,j \in [m+1,m+r] \\ G_{i(j-m)}, & i \in [1,m], j \in [m+1,m+r] \\ G_{(i-m)j}, & i \in [m+1,m+r], j \in [1,m] \end{cases} \tag{3.8}$$

3.2.4 Publisher-News Relation Embedding

Fake news is often written to convey opinions or claims that support the partisan bias of news publishers. Thus, a good news representation should be good at predicting the partisan bias of its publisher. We obtain the list of publishers' partisan scores from a well-known media bias fact-checking websites MBFC [3]. The partisan bias labels are checked with a principled methodology that ensures the reliability and objectivity of the partisan annotations. The labels are categorized into five categories: "left", "left-Center","least-biased","right-Center" and "right". To further ensure the accuracy of the labels, we only consider those news publishers with the annotations ["left","least-biased", "Right"], and rewrite the corresponding labels as [-1,0,1]. Thus, we can construct a partisan label vectors for news publishers as $\mathbf{o}$. Note that we may not obtain the partisan labels for all publishers, so we introduce $\mathbf{e} \in \{0,1\}^{l\times 1}$ to control the weight of $\mathbf{o}$. If we have the partisan bias label of publisher p_k, then $\mathbf{e}_k = 1$; otherwise, $\mathbf{e}_k = 0$. The basic idea is to utilize publisher partisan labels vector $\mathbf{o} \in \mathbb{R}^{l\times 1}$ and publisher-news matrix $\mathbf{B} \in \mathbb{R}^{l\times n}$ to optimize the news feature representation learning. Specifically, we optimization following objective function,

[3]https://mediabiasfactcheck.com/

$$\min_{\mathbf{D}\geq 0,\mathbf{q}} \| \mathbf{e} \odot (\bar{\mathbf{B}}\mathbf{D}\mathbf{q} - \mathbf{o})\|_2^2 + \lambda\|\mathbf{q}\|_2^2 \tag{3.9}$$

where we assume that the latent feature of news publisher can be represented by the features of all the news it published, i.e., $\bar{\mathbf{B}}\mathbf{D}$. $\bar{\mathbf{B}}$ is the normalized user-news publishing relation matrix, i.e., $\bar{\mathbf{B}}_{kj} = \frac{\mathbf{B}_{kj}}{\sum_{j=1}^{n}\mathbf{B}_{kj}}$. $\mathbf{q} \in \mathbb{R}^{d\times 1}$ is the weighting matrix that maps news publishers' latent features to corresponding partisan label vector $\mathbf{o}$.

We have introduced how we can learn news latent features by modeling different aspects of the tri-relationship. We further employ a semi-supervised linear classifier term as follows,

$$\min_{\mathbf{p}} \| \mathbf{D}_L\mathbf{p} - \mathbf{y}_L\|_2^2 + \lambda\|\mathbf{p}\|_2^2 \tag{3.10}$$

where $\mathbf{p} \in \mathbb{R}^{d\times 1}$ is the weighting matrix that maps news latent features to fake news labels. With all previous components, TriFN solves the following optimization problem,

$$\begin{aligned}\min_{\mathbf{D},\mathbf{U},\mathbf{V},\mathbf{T}\geq 0,\mathbf{p},\mathbf{q}} & \|\mathbf{X} - \mathbf{D}\mathbf{V}^T\|_F^2 + \alpha\|\mathbf{Y} \odot (\mathbf{A} - \mathbf{U}\mathbf{T}\mathbf{U}^T)\|_F^2 \\ & + \beta\mathrm{tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) + \gamma\|\mathbf{e} \odot (\bar{\mathbf{B}}\mathbf{D}\mathbf{q} - \mathbf{o})\|_2^2 \\ & + \eta\|\mathbf{D}_L\mathbf{p} - \mathbf{y}_L\|_2^2 + \lambda R\end{aligned} \tag{3.11}$$

where $R = (\|\mathbf{D}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{U}\|_F^2 + \|\mathbf{T}\|_F^2 + \|\mathbf{p}\|_2^2 + \|\mathbf{q}\|_2^2)$ is to avoid over-fitting. The first term models the news latent features from news contents; the second term extracts user latent features from their social relationships; and the third term incorporates the user-news interactions; and the fourth term models publisher-news relationships. The fifth term adds a semi-supervised fake news classifier. Therefore, this framework provides a principled way to model tri-relationship for fake news prediction.

Table 1: The statistics of the datasets

Platform	Users	Engagements	Social Links	Fake News	True News	Publishers
BuzzFeed (**B**)	15,257	25, 240	634,750	91	91	9
PolitiFact (**P**)	23,865	37,259	574,744	120	120	91

3.3 Evaluating TriFN

In this section, we conduct experiments to evaluate the effectiveness of TriFN for fake news detection.

3.3.1 Experimental Settings

We utilize one of the comprehensive fake news detection benchmark dataset called FakeNewsNet (Shu *et al.*, 2017, 2018b). The dataset is collected from two platforms with fact-checking: *BuzzFeed* and *PolitiFact*, both containing news content with labels and social context information. News content includes the meta attributes of the news (e.g., body text), and social context includes the related user social engagements of news items (e.g., user posting/sharing news in Twitter). The detailed statistics of the datasets are shown in Table 8. To evaluate the performance of fake news detection algorithms, we use the following metrics, which are commonly used to evaluate classifiers in related areas: Accuracy (Acc.), Precision (Prec.), Recall (Rec.), and F1. We randomly choose 80% of news pieces for training and remaining 20% for testing, and the process is performed for 10 times and the average performance is reported.

We randomly choose 80% of news pieces for training and remaining 20% for testing, and the process is performed for 10 times and the average performance is

reported. We compare the proposed framework TriFN with several state-of-the-art fake news detection methods. Existing methods mainly focus on extracting *discriminative features* and feed them into a classification algorithm to differentiate fake news. Next, we introduce several representative features as follows,

- **RST** (Rubin *et al.*, 2015): RST stands for Rhetorical Structure Theory, which builds a tree structure to represent rhetorical relations among the words in the text. RST can extract style-based features of news by mapping the frequencies of rhetorical relations to a vector space [4];
- **LIWC** (Pennebaker *et al.*, 2015): LIWC stands for Linguistic Inquiry and Word Count, which is widely used to extract the lexicons falling into psycholinguistic categories. It's based on a large sets of words that represent psycholinguistic processes, summary categories, and part-of-speech categories. It learns a feature vector from a psychology and deception perspective [5];
- **Castillo** (Castillo *et al.*, 2011): Castillo extract various kinds of features from those users who have shared a news item on social media. The features are extracted from user profiles and friendship network. We also include the credibility score of users as an additional social context feature;
- **RST+Castillo**: RST+Castillo represents the concatenated features of RST and Castillo, which include features extracted from both news content and social context;
- **LIWC+Castillo**: LIWC+Castillo represents the concatenated features of

[4]The code is available at: https://github.com/jiyfeng/DPLP

[5]The readers can find more details about the software and feature description at: http://liwc.wpengine.com/

Table 2: Best performance comparison for fake news detection

		RST	LIWC	Castillo	RST+Castillo	LIWC+Castillo	TriFN
BuzzFeed	Acc.	0.600	0.719	0.800	0.816	0.825	**0.864**
	Prec.	0.662	0.722	0.822	0.879	0.821	**0.849**
	Rec.	0.615	0.732	0.776	0.748	0.829	**0.893**
	F1	0.633	0.709	0.797	0.805	0.822	**0.870**
PolitiFact	Acc.	0.604	0.688	0.796	0.838	0.829	**0.878**
	Prec.	0.564	0.725	0.767	0.851	0.821	**0.867**
	Rec.	0.705	0.617	0.889	0.824	0.879	**0.893**
	F1	0.615	0.666	0.822	0.835	0.843	**0.880**

LIWC and Castillo, which consists of feature information from both news content and social context.

Note that for a fair and comprehensive comparison, we choose the above feature extraction methods from following aspects: 1) only extract features from **news contents**, such as RST, LIWC; 2) only construct features from **social context**, such as Castillo; and 3) consider both **news content and social context**, such as RST+Castillo, LIWC+Castillo.

3.3.2 Evaluating Detection Performance

We evaluate the effectiveness of the proposed framework TriFN for fake news classification. We determine model parameters with cross-validation strategy, and we repeat the generating process of training/test set for three times and the average performance is reported. We first perform cross validation on parameters $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$, and choose those parameters that achieves best performance, i.e., $\lambda = 0.1$. We also choose latent dimension $d = 10$ for easy parameter tuning, and focus on the parameters that contribute the tri-relationship modeling components. The parameters for TriFN are set as $\{\alpha = 1e-4, \beta = 1e-5, \gamma = 1, \eta = 1\}$ for BuzzFeed and $\{\alpha = 1e-5, \beta = 1e-4, \gamma = 10, \eta = 1\}$ for PolitiFact.

Table 3: Average F1 of baselines for different learning algorithms on BuzzFeed. Best scores are highlighted.

Method	**RST**	**LIWC**	**Castillo**	**RST +Castillo**	**LIWC +Castillo**
LogReg	0.519	0.660	0.714	0.728	0.760
NBayes	0.511	0.370	0.600	0.716	0.680
DTree	0.566	0.581	0.736	0.681	0.772
RForest	0.538	0.709	0.767	0.805	0.733
XGBoost	0.480	0.672	0.797	0.795	0.782
AdaBoost	0.633	0.701	0.724	0.791	0.768
GradBoost	0.492	0.699	0.772	0.724	0.822

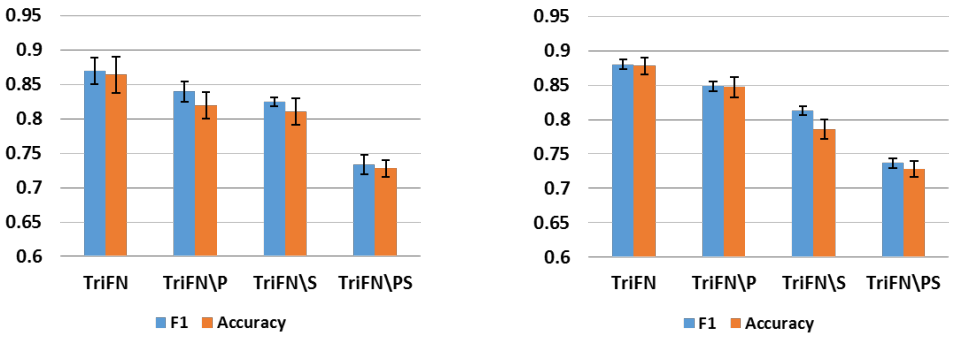
Table 4: Average F1 of baselines for different learning algorithms on PolitiFact. Best scores are highlighted.

Method	**RST**	**LIWC**	**Castillo**	**RST +Castillo**	**LIWC +Castillo**
LogReg	0.615	0.432	0.707	0.668	0.653
NBayes	0.537	0.486	0.442	0.746	0.687
DTree	0.514	0.661	0.771	0.792	0.772
RForest	0.463	0.586	0.767	0.835	0.836
XGBoost	0.552	0.648	0.822	0.783	0.823
AdaBoost	0.502	0.666	0.800	0.787	0.831
GradBoost	0.517	0.650	0.818	0.803	0.843

We test the baseline features on different learning algorithms, and choose the one that achieves the best performance (see Table 9). The algorithms include Logistic Regression (LogReg for short), Naïve Bayes (NBayes), Decision Tree (DTree), Random Forest (RForest), XGBoost, AdaBoost, and Gradient Boosting (GradBoost). We used the open-sourced *xgboost* (Chen and Guestrin, 2016) package and *scikit-learn* (Pedregosa *et al.*, 2011) machine learning framework in Python to implement all these algorithms. To ensure a fair comparison of features, we ran all the algorithms using default parameter settings. We also show the performances for each learning algorithm and report the average performance on both datasets. Due to the space limitation, we

only show the results of F1 score (Table 3 and Table 4). We observe similar results for other metrics in terms of average performance. Based on Table 9, Table 3, and Table 4, we have following observations:

- For news content based methods RST and LIWC, we can see that $LIWC > RST$ for both best performance and average performance, indicating that LIWC can better capture the linguistic features in news contents. The good results of LIWC demonstrate that fake news pieces are very different from real news in terms of choosing the words that reveal psychometrics characteristics.
- In addition, social context based features are more effective than news content based features, i.e., $Castillo > RST$ and $Castillo > LIWC$. It shows that social context features have more discriminative power than those only on news content for predicting fake news.
- Moreover, methods using both news contents and social context perform better than those methods purely based on news contents, and those methods only based on social engagements, i.e., $LIWC + Castillo > LIWC$ or $Castillo$ and $RST + Castillo > RST$ or $Castillo$. This indicates that features extracted from news content and corresponding social context have complementary information, and thus boost the detection performance.
- Generally, for methods based on both news content and social context (i.e., RST+Castillo, LIWC+Castillo, and TriFN), we can see that TriFN consistently outperforms the other two baselines, i.e., $TriFN > LIWC + Castillo$ and $TriFN > RST + Castillo$, in terms of all evaluation metrics on both datasets. For example, TriFN achieves average relative improvement of $4.72\%, 5.84\%$ on BuzzFeed and $5.91\%, 4.39\%$ on PolitiFact, comparing with LIWC+Castillo



(a) BuzzFeed (b) PolitiFact

Figure 3: Impact analysis of users and publishers for fake news detection.

in terms of *Accuracy* and $F1$ score. It supports the importance to model tri-relationship of publisher-news and news-user to better predict fake news.

3.3.3 Assessing Relation Modeling

In previous section, we observe that TriFN framework improves the classification results significantly. In addition to news contents, we also captures user-news interactions and publisher-news relations. Now, we investigate the effects of these components by defining three variants of TriFN:

- TriFN\P - We eliminate the effect of publisher partisan modeling part.
- TriFN\S - We eliminate the effects of user social engagements components.
- TriFN\PS - We eliminate the effects of both publisher partisan and user social engagements.

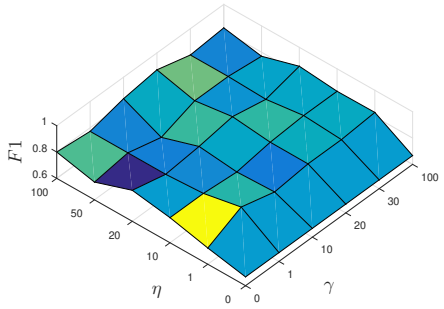
The parameters in all the variants are determined with cross-validation and the best performances are shown in Figure 7, we have following observations:

- When we eliminate the effect of user social engagements component , the performance of TriFN\S degrades in comparison with TriFN. For example, the performance reduces 5.2% and 6.1% in terms of F1 and Accuracy metrics on BuzzFeed, 7.6% and 10.6% on PolitiFact. The results suggest that social engagements in TriFN is important.
- We have similar observations for TriFN\P when eliminating the effect of publisher partisan component. The results suggest the importance to consider publisher-news relations through publisher partisan bias in TriFN.
- When we eliminate both components in TriFN\PS, the results are further reduced compared to TriFN\S and TriFN\P. It also suggests that components of user-news and publisher-news embedding are complementary to each other.

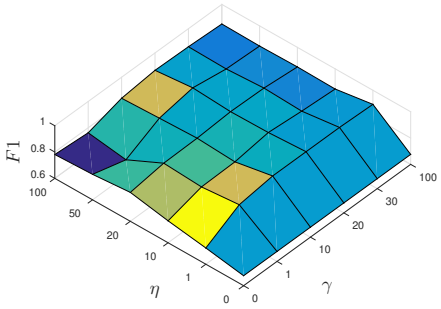
Through the component analysis of TriFN, we conclude that (i) both components can contribute to the performance improvement of TriFN; (ii) it's necessary to model both news contents and social engagements because they contain complementary information.

3.3.4 Parameter Analysis

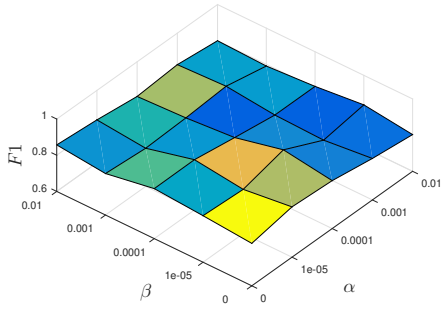
The proposed TriFN has four important parameters. The first two are α and β, which control the contributions from social relationship and user-news engagements. γ controls the contribution of publisher partisan and η controls the contribution of semi-supervised classifier. We first fix $\{\alpha = 1e-4, \beta = 1e-5\}$ and $\{\alpha = 1e-5, \beta = 1e-4\}$ for BuzzFeed and PolitiFact, respectively. Then we vary η as $\{1, 10, 20, 50, 100\}$ and γ in $\{1, 10, 20, 30, 100\}$. The performance variations are depicted in Figure 4. We can



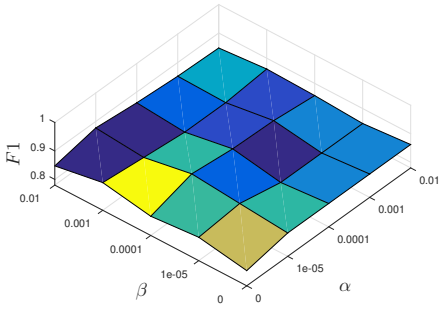
(a) η and γ on BuzzFeed



(b) η and γ on PolitiFact



(c) α and β on BuzzFeed



(d) α and β on PolitiFact

Figure 4: Model parameter analysis for TriFN on BuzzFeed and PolitiFact in terms of F1.

see i) when η increases from 0, eliminating the impact of semi-supervised classification term, to 1, the performance increase dramatically in both datasets. These results support the importance to combine semi-supervised classifier to feature learning; ii) generally, the increase of γ will increase the performance in a certain region, $\gamma \in [1, 50]$ and $\eta \in [1, 50]$ for both datasets, which easy the process for parameter setting. Next, we fix $\{\gamma = 1, \eta = 1\}$ and $\{\gamma = 10, \eta = 1\}$ for BuzzFeed and PolitiFact, respectively. Then we vary $\alpha, \beta \in [0, 1e-5, 1e-4, 1e-3, 0.001, 0.01]$. We can see that i) when α and β increase from 0, which eliminate the social engagements, to $1e-5$, the performance increases relatively, which again support the importance of social engagements; ii)

The performance tends to increase first and then decrease, and it's relatively stable in $[1e-5, 1e-3]$.

Chapter 4

EXPLAINABLE FAKE NEWS DETECTION

In this chapter, we study the explainable fake news detection. Despite the promising results of existing fake news detection methods, however, the majority of these methods focus on *detecting* fake news effectively with latent features but cannot explain "why" a piece of news was detected as fake news. Being able to *explain* why news was determined as fake is much desirable because: (1) the derived explanation can provide new insights and knowledge originally hidden to practitioners; and (2) extracting explainable features from noisy auxiliary information can further help improve fake news detection performance. However, to our best knowledge, there has been no prior attempt to computationally detect fake news with proper explanation.

In particular, we propose to derive explanation from the perspectives of news contents and user comments (See Figure 5). First, news contents may contain

Figure 5: A piece of fake news on PolitiFact, and the user comments on social media. Some explainable comments are directly related to the sentences in news contents.

information that is verifiably false. For example, journalists manually check the claims in news articles on fact-checking websites such as PolitiFact[6], which is usually labor-intensive and time-consuming. Researchers also attempt to use external sources to fact-check the claims in news articles to decide and explain whether a news piece is fake or not (Ciampaglia *et al.*, 2015), which may not be able to check newly emerging events (that has not been fact-checked). Second, user comments have rich information from the crowd on social media, including opinions, stances, and sentiment, that are useful to detect fake news. For example, researchers propose to use social features to select important comments to predict fake news pieces (Guo *et al.*, 2018a). Moreover, news contents and user comments inherently are *related* each other and can provide important cues to explain why a given news article is fake or not. For example, in Figure 5, we can see users discuss different aspects of the news in comments such as "`St. Nicholas was white? Really??Lol`," which directly responds to the claims in the news content "`The Holy Book always said Santa Claus was white`."

In essence, we address the following challenges: (1) How to perform explainable fake news detection that can improve detection performance and explainability simultaneously; (2) How to extract explainable comments without the ground truth during training; and (3) How to model the correlation between news contents and user comments jointly for explainable fake news detection? Next, I will introduce the details of our novel framework named as dEFEND (Explainable FakE News Detection).

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

4.1 Problem Statement

Let a be a news article, consisting of L sentences $\{s_i\}_{i=1}^{L}$. Each sentence $s_i = \{w_1^i, \cdots, w_{M_i}^i\}$ contains M_i words. Let $\mathcal{C} = \{c_1, c_2, ..., c_T\}$ be a set of T comments related to the news a, where each comment $c_j = \{w_1^j, \cdots, w_{Q_j}^j\}$ contains Q_j words. Similar to previous research (Shu *et al.*, 2017; Jin *et al.*, 2016), we treat fake news detection problem as the binary classification problem, i.e., each news article can be true ($y = 1$) or fake ($y = 0$). At the same time, we aim to learn a rank list RS from all sentences in $\{s_i\}_{i=1}^{L}$, and a rank list RC from all comments in $\{c_j\}_{j=1}^{T}$, according to the degree of explainability, where RS_k (RC_k) denotes the k_{th} most explainable sentence (comment). The explainability of sentences in news contents represent the degree of how check-worthy they are, while the explainability of comments denote the degree of how much users believe if news is fake or real, closely related to the major claims in news. Formally, we can represent the problem as follows.

Problem Statement: Given a news article a and a set of related comments $\mathcal{C}$, learn a fake news detection function f: $(\hat{y}, RS, RC) \rightarrow f(A, C)$, such that it maximizes prediction accuracy with explainable sentences and comments ranked highest in RS and RC respectively.

4.2 The Proposed Framework dEFEND

Now, we present the details of the framework for explainability fake news detection, named as dEFEND (neural Explainable FakE News Detection). It consists of four major components (see Figure 6): (1) a news content encoder (including word encoder and sentence encoder) component, (2) a user comment encoder component, (3) a sentence-comment co-attention component, and (4) a fake news prediction component.

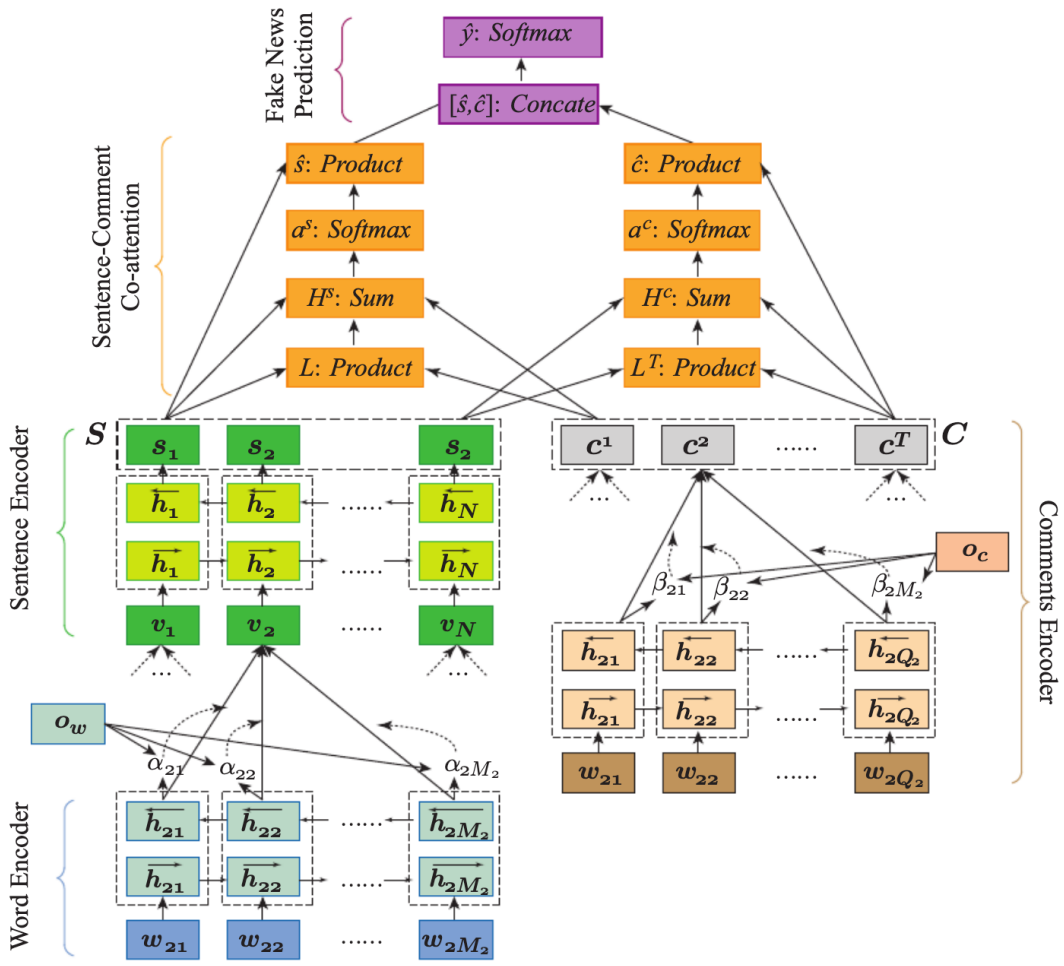


Figure 6: The proposed framework dEFEND consists of four components: (1) a news content (including word-level and sentence-level) encoder, (2) a user comment encoder, (3) a sentence-comment co-attention component, and (4) a fake news prediction component.

4.2.1 Capturing Hierarchical News Structure

As fake news pieces are intentionally created to spread inaccurate information rather than to report objective claims, they often have opinionated and sensational language styles, which have the potential to help detect fake news. In addition, a news document contains linguistic cues with different levels such as word-level and sentence-

level, which provide different degrees of importance for the explainability of why the news is fake. For example, in a fake news claim "`Pence: Michelle Obama is the most vulgar first lady we've ever had`", the word "`vulgar`" contributes more signals to decide whether the news claim is fake rather than other words in the sentence.

Word Encoder We learn the sentence representation via a recurrent neural network (RNN) based word encoder with Gated recurrent units (GRU). To further capture the contextual information of annotations, we use bidirectional GRU (Bahdanau *et al.*, 2016) to model word sequences from both directions of words. The bidirectional GRU contains the forward GRU $\overrightarrow{f}$ which reads sentence s_i from word w_{i1} to w_{iM_i} and a backward GRU $\overleftarrow{f}$ which reads sentence s_i from word w_{iM_i} to w_{i1}:

$$\overrightarrow{\mathbf{h}_{it}} = \overrightarrow{GRU}(\mathbf{w}_{it}), t \in \{1, \ldots, M_i\}, \qquad \overleftarrow{\mathbf{h}_{it}} = \overleftarrow{GRU}(\mathbf{w}_{it}), t \in \{M_i, \ldots, 1\} \tag{4.1}$$

We then obtain an annotation of word w_{it} by concatenating the forward hidden state $\overrightarrow{\mathbf{h}_{it}}$ and backward hidden state $\overleftarrow{\mathbf{h}_{it}}$, i.e., $\mathbf{h}_{it} = [\overrightarrow{\mathbf{h}_{it}}, \overleftarrow{\mathbf{h}_{it}}]$, which contains the information of the whole sentence centered around w_{it}. Note that not all words contribute equally to the representation of the sentence meaning. Therefore, we introduce an attention mechanism to learn the weights to measure the importance of each word, and the sentence vector $\mathbf{v}_i \in \mathbb{R}^{2d \times 1}$ is computed as $\mathbf{v}_i = \sum_{t=1}^{M_i} \alpha_{it} \mathbf{h}_{it}$, where α_{it} measures the importance of t^{th} word for the sentence s_i, and α_{it} is calculated as follows,

$$\mathbf{o}_{it} = \tanh(\mathbf{W}_w \mathbf{h}_{it} + \mathbf{b}_w), \qquad \alpha_{it} = \frac{\exp(\mathbf{o}_{it} \mathbf{o}_w^\mathsf{T})}{\sum_{k=1}^{M_i} \exp(\mathbf{o}_{ik} \mathbf{o}_w^\mathsf{T})} \tag{4.2}$$

where $\mathbf{o}_{it}$ is a hidden representation of $\mathbf{h}_{it}$ obtained by feeding the hidden state $\mathbf{h}_{it}$ to a fully embedding layer, and $\mathbf{o}_w$ is the weight parameter that represents the world-level context vector.

Sentence Encoder Similar to word encoder, we utilize RNNs with GRU units to encode each sentence in news articles. Through the sentence encoder, we can capture the context information in the sentence-level to learn the sentence representations $\mathbf{h}_i$ from the learned sentence vector $\mathbf{v}_i$. Specifically, we can use the bidirectional GRU to encode the sentences as follows:

$$\overrightarrow{\mathbf{h}_i} = \overrightarrow{GRU}(\mathbf{v}_i), i \in \{1, \ldots, L\}, \quad \overleftarrow{\mathbf{h}_i} = \overleftarrow{GRU}(\mathbf{v}_i), i \in \{L, \ldots, 1\} \tag{4.3}$$

We then obtain an annotation of sentence $\mathbf{s}_i \in \mathbb{R}^{2d \times 1}$ by concatenating the forward hidden state $\overrightarrow{\mathbf{h}_i}$ and backward hidden state $\overleftarrow{\mathbf{h}_i}$, i.e., $\mathbf{s}_i = [\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_i}]$, which captures the context from neighbor sentences around sentence s_i.

4.2.2 Encoding User Comments

People express their emotions or opinions towards fake news through social media posts such as comments, such as skeptical opinions, sensational reactions, etc. These textual information has been shown to be related to the content of original news pieces. Thus, comments may contain useful semantic information that has the potential to help fake news detection. Next, we demonstrate how to encode the comments to learn the latent representations. The comments extracted from social media are usually short text, so we use RNNs to encode the word sequence in comments directly to learn the latent representations of comments. Similar to the word encoder, we adopt bidirectional GRU to model the word sequences in comments. Specifically, given a comment c_j with words $w_{jt}, t \in \{1, \cdots, Q_j\}$, we first map each word w_{jt} into the word vector $\mathbf{w}_{jt} \in \mathbb{R}^d$ with an embedding matrix. Then, we can obtain the feed forward hidden states $\overrightarrow{\mathbf{h}_{jt}}$ and backward hidden states $\overleftarrow{\mathbf{h}_{jt}}$ as follows,

$$\overrightarrow{\mathbf{h}_{jt}} = \overrightarrow{GRU}(\mathbf{w}_{jt}), t \in \{1, \ldots, Q_j\}, \quad \overleftarrow{\mathbf{h}_{jt}} = \overleftarrow{GRU}(\mathbf{w}_{jt}), t \in \{Q_j, \ldots, 1\} \tag{4.4}$$

We further obtain the annotation of word w_{jt} by concatenating $\overrightarrow{\mathbf{h}_{jt}}$ and $\overleftarrow{\mathbf{h}_{jt}}$, i.e., $\mathbf{h}_{jt} = [\overrightarrow{\mathbf{h}_{jt}}, \overleftarrow{\mathbf{h}_{jt}}]$. We also introduce the attention mechanism to learn the weights to measure the importance of each word, and the comment vector $\mathbf{c}_j \in \mathbb{R}^{2d}$ is computed as $\mathbf{c}_j = \sum_{t=1}^{Q_j} \beta_{jt} \mathbf{h}_{jt}$, where β_{jt} measures the importance of t^{th} word for the comment c_j, and β_{jt} is calculated as follows,

$$\mathbf{o}_{jt} = \tanh(\mathbf{W}_c \mathbf{h}_{jt} + \mathbf{b}_c), \qquad \beta_{jt} = \frac{\exp(\mathbf{o}_{jt} \mathbf{o}_c^{\mathsf{T}})}{\sum_{k=1}^{Q_j} \exp(\mathbf{o}_k^j \mathbf{o}_c^{\mathsf{T}})} \tag{4.5}$$

where $\mathbf{o}_{jt}$ is a hidden representation of $\mathbf{h}_{jt}$ obtained by feeding the hidden state $\mathbf{h}_{jt}$ to a fully embedding layer, and $\mathbf{u}_c$ is the weight parameter.

4.2.3 Sentence-Comment Co-attentions

We observe that not all sentences in news contents are fake, and in fact, many sentences are true but only for supporting wrong claim sentences (Feng *et al.*, 2012). Thus, news sentences may not be equally important in determining and explaining whether a piece of news is fake or not. Similarly, user comments may contain relevant information about the important aspects that explain why a piece of news is fake, while they may also be less informative and noisy.

Thus, we aim to select some news sentences and user comments that can explain why a piece of news is fake. As they provide a good explanation, they should also be helpful in detecting fake news. This suggests us to design attention mechanisms to give high weights of representations of news sentences and comments that are beneficial to fake news detection. Specifically, we use sentence-comment co-attention because it can capture the semantic affinity of sentences and comments and further help learn the attention weights of sentences and comments simultaneously.

We can construct the feature matrix of news sentences $\mathbf{S} = [\mathbf{s}_1; \cdots, \mathbf{s}_L] \in \mathbb{R}^{2d \times L}$

and the feature map of user comments $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_T\} \in \mathbb{R}^{2d \times T}$, the co-attention attends to the sentences and comments simultaneously. Similar to (Lu *et al.*, 2016), we first compute the affinity matrix $\mathbf{L} \in \mathbb{R}^{T \times L}$ as follows,

$$\mathbf{L} = \tanh(\mathbf{C}^{\mathsf{T}} \mathbf{W}_l \mathbf{S}) \tag{4.6}$$

where $\mathbf{W}_l \in \mathbb{R}^{2d \times 2d}$ is a weight matrix to be learned through the networks. Following the optimization strategy in (Lu *et al.*, 2016), we can consider the affinity matrix as a feature and learn to predict sentence and comment attention maps as follows,

$$\mathbf{H}^s = \tanh(\mathbf{W}_s \mathbf{S} + (\mathbf{W}_c \mathbf{C})\mathbf{L}), \quad \mathbf{H}^c = \tanh(\mathbf{W}_c \mathbf{C} + (\mathbf{W}_s \mathbf{S})\mathbf{L}^{\mathsf{T}}) \tag{4.7}$$

where $\mathbf{W}_s, \mathbf{W}_c \in \mathbb{R}^{k \times 2d}$ are the weight parameters. The attention weights of sentences and comments are calculated as follows,

$$\mathbf{a}^s = softmax(\mathbf{w}_{hs}^{\mathsf{T}} \mathbf{H}^s), \quad \mathbf{a}^c = softmax(\mathbf{w}_{hc}^{\mathsf{T}} \mathbf{H}^c) \tag{4.8}$$

where $\mathbf{a}^s \in \mathbb{R}^{1 \times N}$ and $\mathbf{a}^c \in \mathbb{R}^{1 \times T}$ are the attention probabilities of each sentence $\mathbf{s}_i$ and comment $\mathbf{c}^j$, respectively. $\mathbf{w}_{hs}, \mathbf{w}_{hc} \in \mathbb{R}^{1 \times k}$ are the weight parameters. The affinity matrix $\mathbf{F}$ transforms user comment attention space to news sentence attention space, and vice versa for $\mathbf{F}^{\mathsf{T}}$. Based on the above attention weights, the comment and sentence attention vectors are calculated as the weighted sum of the comment features and sentence features, i.e., $\hat{\mathbf{s}} = \sum_{i=1}^{L} \mathbf{a}_i^s \mathbf{s}_i$ and $\hat{\mathbf{c}} = \sum_{j=1}^{T} \mathbf{a}_j^c \mathbf{c}_j$, where $\hat{\mathbf{s}} \in \mathbb{R}^{1 \times 2d}$ and $\hat{\mathbf{c}} \in \mathbb{R}^{1 \times 2d}$ are the learned features for news sentences and user comments through co-attention.

We have introduced how we can encode news contents by modeling the hierarchical structure from word level and sentence level, how we encode comments by word-level attention networks, and the component to model co-attention to learn sentences and comments representations. We further integrate these components together and

Table 5: The statistics of the datasets

Platform	Users	Comments	True News	Fake News
PolitiFact (**P**)	68,523	89,999	145	270
Gossipcop (**G**)	156,467	231,269	3,586	2,230

predict fake news with the following objective,

$$\hat{\mathbf{y}} = \text{softmax}([\hat{\mathbf{s}}, \hat{\mathbf{c}}]\mathbf{W}_f + \mathbf{b}_f) \tag{4.9}$$

where $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_1]$ is the predicted probability vector with $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_1$ indicate the predicted probability of label being 0 (real news) and 1 (fake news) respectively. $y \in \{0, 1\}$ denotes the ground truth label of news. $[\hat{\mathbf{s}}, \hat{\mathbf{c}}]$ means the concatenation of learned features for news sentences and user comments. $\mathbf{b}_f \in \mathbb{R}^{1\times 2}$ is the bias term. Thus, for each news piece, the goal is to minimize the cross-entropy loss function as follows,

$$\mathcal{L}(\theta) = -y \log(\hat{\mathbf{y}}_1) - (1 - y) \log(1 - \hat{\mathbf{y}}_1) \tag{4.10}$$

where θ denotes the parameters of the network.

4.3 Evaluating dEFEND

In this section, we evaluate the performance of dEFEND for fake news detection and explainability discovery. Specifically, we aim to answer the following evaluation questions:

- **EQ1** Can dEFEND improve fake news classification performance by modeling news contents and user comments simultaneously?

- **EQ2** How effective are news contents and user comments, respectively, in improving the detection performance of dEFEND?
- **EQ3** Can dEFEND capture the news sentences and user comments that can explain why a piece of news is fake?

4.3.1 Experimental Settings

We utilize the datasets collected from two platforms with fact-checking: *GossipCop* and *PolitiFact*, both containing news content with labels and social context information. News content includes the meta attributes of the news (e.g., body text), and social context includes the related user social engagements of news items (e.g., user comments in Twitter). Note that we keep news pieces with at least 3 comments. The statistics of the datasets are shown in Table 5.

The representative state-of-the-art fake news detection algorithms are listed as follows:

- **RST** (Rubin *et al.*, 2015): RST stands for Rhetorical Structure Theory, which builds a tree structure to represent rhetorical relations among the words in the text. RST can extract news style features by mapping the frequencies of rhetorical relations to a vector space[7];
- **LIWC** (Pennebaker *et al.*, 2015): LIWC stands for Linguistic Inquiry and Word Count, which is widely used to extract the lexicons falling into psycho-linguistic categories. It's based on a large set of words that represent psycho-linguistic

[7]The code is available at https://github.com/jiyfeng/DPLP

processes, summary categories, and part-of-speech categories. It learns a feature vector from psychology and deception perspective[8];

- **LIWC** (Pennebaker *et al.*, 2015): LIWC stands for Linguistic Inquiry and Word Count, which is widely used to extract the lexicons falling into psycho-linguistic categories. It's based on a large set of words that represent psycho-linguistic processes, summary categories, and part-of-speech categories. It learns a feature vector from psychology and deception perspective[9];
- **text-CNN** (Kim, 2014): text-CNN utilizes convolutional neural networks to model news contents, which can capture different granularity of text features with multiple convolution filters;
- **TCNN-URG** (Qian *et al.*, 2018): TCNN-URG consists of two major components: a two-level convolutional neural network to learn representations from news content, and a conditional variational auto-encoder to capture features from user comments;
- **HPA-BLSTM** (Guo *et al.*, 2018a): HPA-BLSTM is a neural network model that learns news representation through a hierarchical attention network on word-level, post-level, and sub-event level of user engagements on social media. In addition, post features are extracted to learn the attention weights during post-level;
- **CSI** (Ruchansky *et al.*, 2017); CSI is a hybrid deep learning model that utilizes information from text, response, and source. The news representation is modeled via an LSTM neural network with the Doc2Vec (Le and Mikolov, 2014)

[8]The readers can find more details about the software and feature description at http://liwc.wpengine.com/

[9]The readers can find more details about the software and feature description at http://liwc.wpengine.com/

Table 6: The performance comparison for fake news detection.

		RST	LIWC	CNN	HAN	TCNN-URG	HPA-BLSTM	CSI	dEFEND
P	Acc.	0.607	0.769	0.653	0.837	0.712	0.846	0.827	**0.904**
	Prec.	0.625	0.843	0.678	0.824	0.711	0.894	0.847	**0.902**
	Rec.	0.523	0.794	0.863	0.896	0.941	0.868	0.897	**0.956**
	F1	0.569	0.818	0.760	0.860	0.810	0.881	0.871	**0.928**
G	Acc.	0.531	0.736	0.739	0.742	0.736	0.753	0.772	**0.808**
	Prec.	0.534	**0.756**	0.707	0.655	0.715	0.684	0.732	0.729
	Rec.	0.492	0.461	0.477	0.689	0.521	0.662	0.638	**0.782**
	F1	0.512	0.572	0.569	0.672	0.603	0.673	0.682	**0.755**

embedding on the news contents and user comments as input, and for a fair comparison, the user features are ignored.

4.3.2 Evaluating Detection Performance

To answer **EQ1**, we first compare dEFEND with representative fake news detection algorithms. We randomly choose 75% of news pieces for training and remaining 25% for testing, and the process is performed for 5 times and the average performance is reported in Table 6. From the table, we make the following observations:

- For news content based methods RST, LIWC and HAN, we can see that $HAN > LIWC > RST$ for both datasets. It indicates that 1) HAN can better capture the syntactic and semantic cues through hierarchical attention neural networks in news contents to differentiate fake and real news; 2) LIWC can better capture the linguistic features in news contents. The good results of LIWC demonstrate that fake news pieces are very different from real news in terms of choosing the words that reveal psychometrics characteristics.
- In addition, methods using both news contents and user comments perform better than those methods purely based on news contents, and those methods

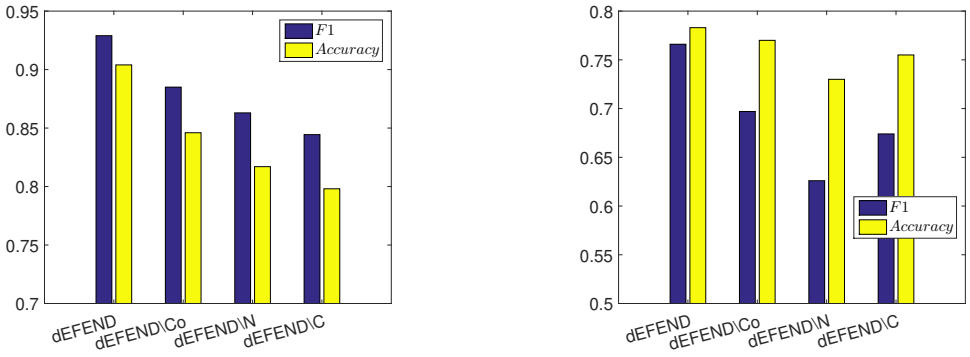
only based on user comments, i.e., $dEFEND > HAN$ or $HPA-BLSTM$ and $CSI > HAN$ or $HPA-BLSTM$. This indicates that features extracted from news content and corresponding user comments have complementary information, and thus boost the detection performance.

- Moreover, the performance of user comment based methods are slightly better than news content based methods. For example, we have $HPA-BLSTM > HAN$ in terms of Accuracy and F1 on both PolitiFact and Gossipcop data. It shows that features extracted from user comments have more discriminative power than those only on news content for predicting fake news.
- Generally, for methods based on both news content and user comments (i.e., dEFEND, CSI, and $TCNN-URG$), we can see that dEFEND consistently outperforms CSI and $TCNN-URG$ and, i.e., $dEFEND > CSI > TCNN-URG$, in terms of all evaluation metrics on both datasets. For example, dEFEND achieves average relative improvement of $4.5\%, 3.6\%$ on PolitiFact and $4.7\%, 10.7\%$ on Gossipcop, comparing with CSI in terms of $Accuracy$ and $F1$ score. It supports the importance of modeling co-attention of news sentences and user comments for fake news detection.

4.3.3 Assessing Impacts of News Contents and User Comments

In addition to news contents, we also capture information from user comments and integrate it with news contents with co-attention. In order to answer **EQ2**, we further investigate the effects of these components by defining three variants of dEFEND:

- **dEFEND\C**: dEFEND\C is a variant of dEFEND without considering infor-



(a) PolitiFact (b) GossipCop

Figure 7: Impact analysis of news contents, comments, and sentence-comment co-attention for fake news detection.

mation from user comments. It first encodes news contents with word-level attentions on each sentence, and then the resultant sentence features are averaged through an average pooling layer and feed into a softmax layer for classification.

- **dEFEND\N**: dEFEND\N is a variant of dEFEND without considering information from news contents. It first utilizes the comment encoder to learn comment features, and then the resultant comment features are averaged through an average pooling layer and feed into a softmax layer for classification.
- **dEFEND\Co**:dEFEND\Co is a variant of dEFEND, which eliminates the sentence-comment co-attention. Instead, it performs self-attention on sentences and comments separately and the resultant features are concatenated to a dense layer and feed into a softmax layer for classification.

The parameters in all the variants are determined with cross-validation and the best performances are reported in Figure 7. We make the following observations:

- When we eliminate the co-attention for news contents and user comments, the performances are reduced. It suggests the importance of modeling the correlation and captures the mutual influence between news contents and user comments.

- When we eliminate the effect of news contents, the performance of dEFEND\N degrades in comparison with dEFEND. For example, the performance reduces 4.2% and 6.6% in terms of F1 and Accuracy metrics on PolitiFact, 18.2% and 6.8% on GossipCop. The results suggest that news contents in dEFEND are important.
- We have a similar observation for dEFEND\C when eliminating the effect of user comments. The results suggest the importance to consider the feature of user comments to guide fake news detection in dEFEND.

Through the component analysis of dEFEND, we conclude that (1) both components of news contents and user comments can contribute to the fake news detection performance improvement of dEFEND; (2) it is necessary to model both news contents and user comments because they contain complementary information.

4.3.4 Explainability Evaluation

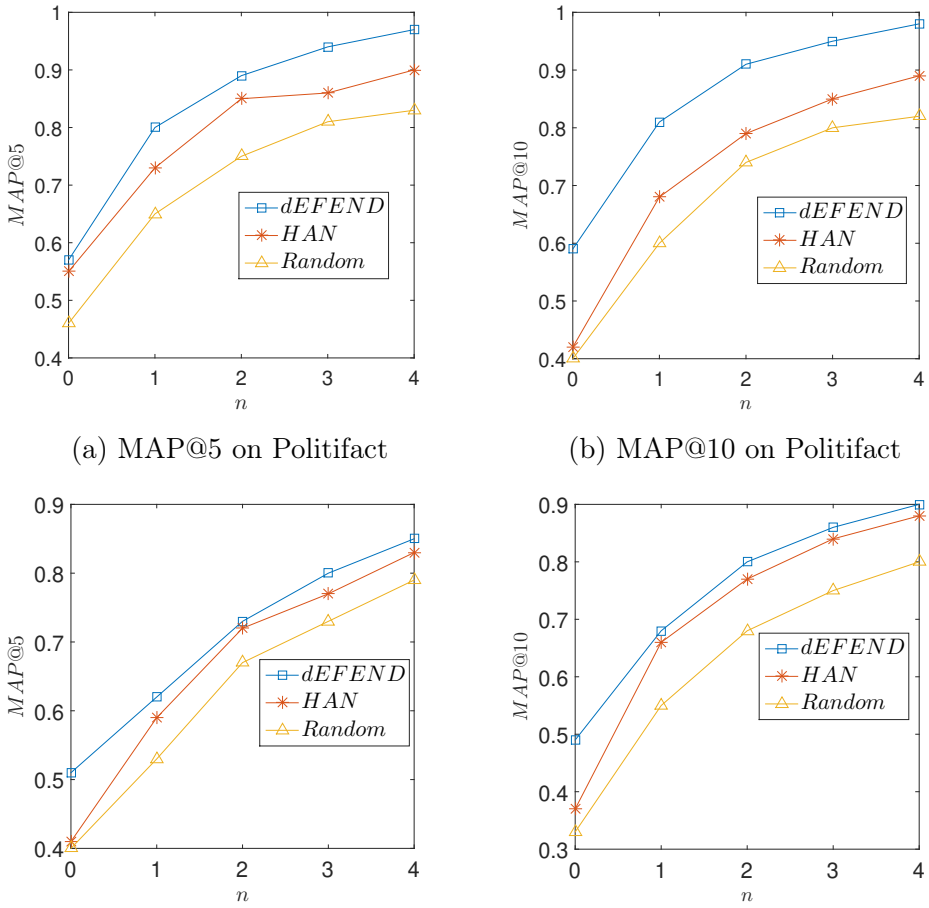
In this subsection, we evaluate the performance of explainability of dEFEND framework from the perspective of news sentences and user comments. It is worth mentioning that all of the baseline methods are designed for fake news detection, and none of them are initially proposed to discover explainable news sentences or user comments.

News Sentence Explainability: We aim to demonstrate the performance of the explainability rank list of news sentences, i.e., RS. Specifically, we want to see if the top-ranked explainable sentences determined by our method are more likely to be related to the major claims in fake news that are worth to check–i.e., check-worthy. Therefore, we utilize ClaimBuster (Hassan *et al.*, 2017) to obtain a ground truth rank

list $\tilde{RS}$ of all check-worthy sentences in a piece of news content. ClaimBuster proposes a scoring model that utilizes various linguistics features trained using tens of thousands of sentences from past general election debates that were labeled by human coders and gives a "check-worthiness" score between 0 and 1. The higher the score, the more likely the sentence contains check-worthy factual claims. The lower the score, the more non-factual, subjective and opinionated the sentence is. We compare top-k rank list of the explainable sentences in news contents by dEFEND ($RS^{(1)}$) and HAN ($RS^{(2)}$), with top-k rank list, $\tilde{RS}$, by ClaimBuster, using the evaluation metric, MAP@k (Mean Average Precision), where k is set as 5 and 10. We also introduce another parameter n which controls the window size that allows n neighboring sentences are considered when comparing the sentences in $RS^{(1)}$ and $RS^{(2)}$ with each of the top-k sentences in $\tilde{RS}$. From Figure 8, we make the following observations:

- In general, we can see that $dEFEND > HAN > Random$ for the performance of finding check-worthy sentences in news contents on both datasets. It indicates that the sentence-comment co-attention component in dEFEND can help selecting more check-worthy sentences.
- With the increase of n, we relax the condition to match check-worthy sentences in the ground truth, and thus the MAP performance is increasing.
- When $n = 1$, the performance of dEFEND on MAP@5 and MAP@10 increases to exceed 0.8 for PolitiFact, which indicates that dEFEND can detect check-worthy sentences well within 1 neighboring sentence of the ground truth sentences in $\tilde{RS}$.

User Comments Explainability: We deploy several tasks using Amazon Mechani-



(a) MAP@5 on Politifact

(b) MAP@10 on Politifact

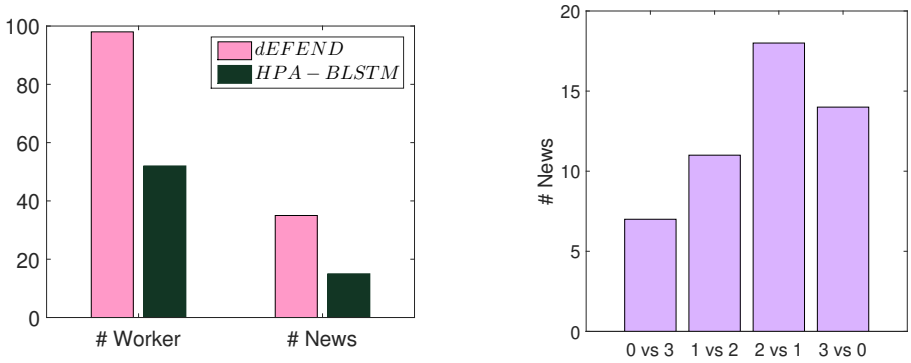
(c) MAP@5 on Gossipcop

(d) MAP@10 on Gossipcop

Figure 8: The performance of sentence explainability on MAP@5 and MAP@10 w.r.t. the neighborhood threshold n.

cal Turk (AMT)[10] to evaluate the explainability rank list of the comments RC for fake news. We perform the following settings to deploy AMT tasks for a total of 50 fake news pieces. To evaluate the explainability of user comments, for each news article, we have two lists of top-k comments, $L^{(1)} = (L_1^{(1)}, L_2^{(1)}, \cdots, L_k^{(1)})$ for using dEFEND and $L^{(2)} = (L_1^{(2)}, L_2^{(2)}, \cdots, L_k^{(2)})$ for HPA-BLSTM. The top-k comments are selected and ranked using the attention weights from the high to low. To evaluate the model

[10]https://www.mturk.com/



(a) Winning Count

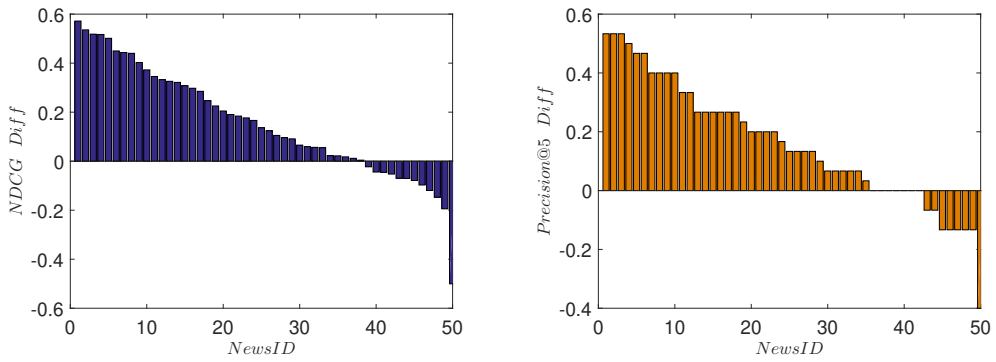
(b) Worker Voting Ratio

Figure 9: The human-evaluation of explainable comment list of dEFEND and HPA-BLSTM.

ability to select topmost explainable comments, we empirically set $k = 5$. We deploy two AMT tasks to evaluate the explainable ranking performance.

For **Task 1**, we perform **list-wise** comparison. We ask workers pick a *collectively* better list between $L^{(1)}$ and $L^{(2)}$. To remove the position bias, we randomly assign the position, top and bottom, of $L^{(1)}$ and $L^{(2)}$ when presented to workers. We let each worker pick the better list between $L^{(1)}$ and $L^{(2)}$ for each news piece. We ensure each news piece is evaluated by 3 workers, and finally obtained 150 results of workers' choices. In a worker-level, we compute the number of workers that choose $L^{(1)}$ and $L^{(2)}$, and also compute the winning ratio (WR for short) for them. In a news-level, we perform majority voting for all 3 workers for each news and decide if workers choose $L^{(1)}$ or $L^{(2)}$. For each news, we also compute the worker-level choices by computing the ratio between $L^{(1)}$ and $L^{(2)}$. From Figure 9, we make the following observations:

- dEFEND can select better top-k explainable comments than HPA-BLSTM both in worker-level and news-level. First, in worker-level, 98 out of 150 workers (with WR=0.65) choose $L^{(1)}$ over $L^{(2)}$. Second, in news-level, dEFEND has better performance in 32 out of 50 news pieces (with WR=0.64) than HPA-BLSTM.



(a) NDCG (b) precision@5

Figure 10: The discrepancy histograms of mean NDCG and mean Precision@5 of the results between two methods.

- We can see that there are more news pieces such that 3 workers vote unanimously for $L^{(1)}$ (3 vs 0) than the opposite case (0 vs 3) for their explainability i.e., 14>7. Similarly, there are more cases where 2 workers vote for dEFEND than HPA-BLSTM, i.e., 18>11.

For **Task 2**, we perform **item-wise** evaluation. For each comment in $L^{(1)}$ and $L^{(2)}$, we ask workers to choose a score from $\{0, 1, 2, 3, 4\}$, where 0 means "not explainable at all," 1 means "not explainable," 3 means "somewhat explainable," 4 means "highly explainable," and 2 means "somewhere in between." To avoid the bias caused by different user criteria, we shuffle the order of comments in $L^{(1)}$ and $L^{(2)}$, and ask workers to assess how explainable each comment is with respect to the news. To estimate rank-aware explainability of comments (i.e., having a higher ranked explainable comment is more desirable than a lower ranked one), we use NDCG (Normalized Cumulative Gain) (Järvelin and Kekäläinen, 2002) and Precision@k as the evaluation metrics. NDCG is widely used in information retrieval to measure document ranking performance in search engines. It can measure how good a ranking is by comparing the

proposed ranking with the ideal ranking list measured by user feedback. Precision@k is the proportion of recommended items in a top-k set that are relevant. Similarly, we ensure each news piece is evaluated by 3 workers and obtain a total of 750 results of workers' ratings for each method. The results are shown in Figure 10, where news articles are sorted by the discrepancy in the metrics between the two methods in descending order (e.g., NDCG(dEFEND)- NDCG(HPA-BLSTM)). We show only the results of Precision@5 as those of Precision@10 are similar. We have the following observations:

- Among 50 fake news articles, dEFEND obtains higher NDCG scores than HPA-BLSRM for 38 cases in terms of the item-wise evaluation. Overall mean NDCG scores over 50 cases for dEFEND and HPA-BLSRM are 0.71 and 0.55, respectively.
- Similar results can be found on Precision@5. dEFEND is superior to HPA-BLSTM on 35 fake news articles and tied on 7 articles. Overall mean Precision@5 scores over 50 cases for dEFEND and HPA-BLSRM are 0.67 and 0.51, respectively.

Case Study: We compare dEFEND with HPA-BLSTM and demonstrate the explainable comments that we correctly ranked high but missed by HPA-BLSTM as in Figure 11. We can see that: (1) dEFEND can rank more explainable comments higher than non-explainable comments. For example, comment "`...president does not have the power to give citizenship...`" is ranked at the top, which can explain exactly why the sentence "`granted U.S. citizenship to 2500 Iranians including family members of government officials`" in the news content is fake; (2) we can give higher weights to explainable comments than those interfering

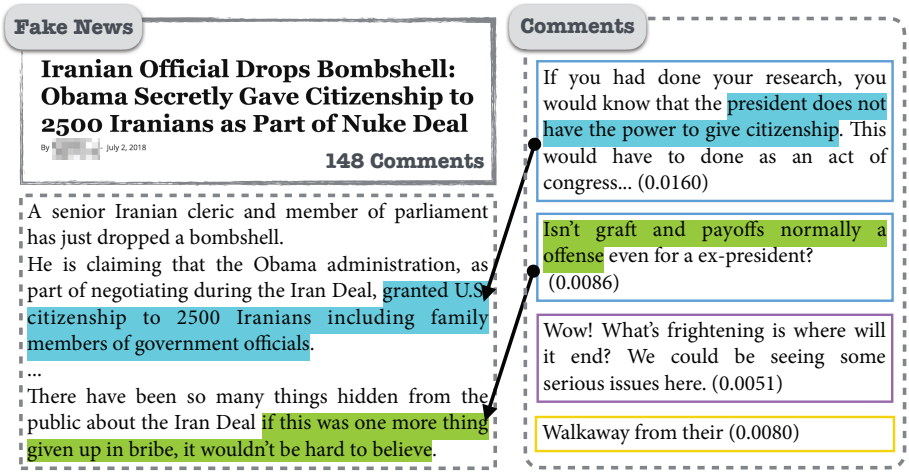


Figure 11: The explainable comments captured by dEFEND.

and unrelated comments, which can help select more related comments to help detect fake news. For example, unrelated comment "`Walkaway from their...`" has an attention weight 0.0080, which is less than an explainable comment "`Isn't graft and payoffs normally a offense`" with an attention weight 0.0086, so the latter comment is selected to be a more important feature for fake news prediction.

Chapter 5

EARLY FAKE NEWS DETECTION

In this chapter, we study early fake news detection. Research has shown that fake news spreads farther, faster, deeper, and more widely than true news (Vosoughi *et al.*, 2018). Widespread fake news can erode the public trust in government and professional journalism and lead to adverse real-life events. Thus, a timely detection of fake news on social media is critical to cultivate a healthy news ecosystem.

It presents unique challenges and opportunities for early detection of fake news. First, fake news is diverse in terms of topics, content, publishing methods and media platforms, and sophisticated linguistic styles geared to emulate true news. Consequently, training machine learning models on such sophisticated content requires *large-scale annotated fake news data* that is egregiously difficult to obtain. Second, it is important to detect fake news early. Most of the research on fake news detection rely on signals that require a long time to aggregate, making them unsuitable for *early detection*. Third, the evolving nature of fake news makes it essential to analyze it with signals from multiple sources to better understand the context. A system solely relying on social networks and user engagements can be easily influenced by biased user feedback, whereas relying only on the content misses the rich auxiliary information from the available sources.

Prior works on detecting fake news (Qian *et al.*, 2018; Wang *et al.*, 2018) rely on large amounts of labeled instances to train supervised models. Such large labeled training data is difficult to obtain in the early phase of fake news detection. To overcome this challenge, learning with weak supervision presents a viable solution.

Figure 12: An illustration of a piece of fake news and related user comments, which can be used for extracting weak social supervision for early detection. Users have different credibility, perceived bias, and express diverse sentiment to the news.

Consider the example in Figure 14. Although it is difficult to determine the veracity considering the news content in isolation, the surrounding context from other users' posts and comments provide clues, in the form of opinions, stances, and sentiment, useful to detect fake news. For example, in Figure 14, the phrase "`kinda agree..`" indicates a positive sentiment to the news, whereas the phrase "`I just do not believe it...`" expresses a negative sentiment. Prior work has shown conflicting sentiments among propagators to indicate a higher probability of fake news (Jin *et al.*, 2016; Shu *et al.*, 2017). Also, users have different credibility degrees in social media and less-credible ones are more likely to share fake news (Shu *et al.*, 2019c). Although we do not have this information a priori, we can consider *agreement* between users as a weak proxy for their credibility. All of the aforementioned signals from different sources of social engagements can be leveraged as weak supervision signals to train machine learning models.

We leverage weak social supervision to detect fake news from limited annotated data. We provide a principled solution, dubbed MWSS to learn from Multiple-sources

of Weak Social Supervision (MWSS) from multi-faceted social media data. Our framework is powered by meta learning with a Label Weighting Network (LWN) to capture the relative contribution of different weak social supervision signals for training; In particular, our model leverages a small amount of manually-annotated clean data and a large amount of weakly annotated data by proxy signals from multiple sources for joint training in a meta-learning framework. Since not all weak instances are equally informative, the model learns to estimate their respective contributions for the end task. The framework is uniquely suitable for early fake news detection, because it (1) leverages rich weak social supervision to boost model learning in a meta-learning fashion; and (2) only requires the news content during the prediction stage without relying on the social context as features for early prediction. Next, I will introduce the details of our novel framework named as MWSS.

5.1 Problem Statement

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ denote a set of n news articles with manually annotated clean labels, with $\mathcal{X} = \{x_i\}_{i=1}^n$ denoting the news pieces and $\mathcal{Y} = \{y_i\}_{i=1}^n \subset \{0,1\}^n$ the corresponding clean labels of whether the news is fake or not. In addition, there is a large set of unlabeled examples. Usually the size of the clean labeled set n is smaller than the unlabeled set due to labeling costs. For the widely available unlabeled samples, we can generate weak labels by using different labeling functions based on *social engagements*. For a specific labeling function $g^{(k)} : \mathcal{X}^{(k)} \rightarrow \tilde{\mathcal{Y}}^{(k)}$, where $\mathcal{X}^{(k)} = \{x_j^{(k)}\}_{j=1}^N$ denotes the set of N unlabeled messages to which the labeling function $g^{(k)}$ is applied and $\tilde{\mathcal{Y}}^{(k)} = \{\tilde{y}_j^{(k)}\}_{j=1}^N$ as the resulting set of weak labels. This

weakly labeled data is denoted by $\tilde{\mathcal{D}}^{(k)} = \{x_j^{(k)}, \tilde{y}_j^{(k)}\}_{j=1}^{N}$ and often $n \ll N$. We formally define our problem as:

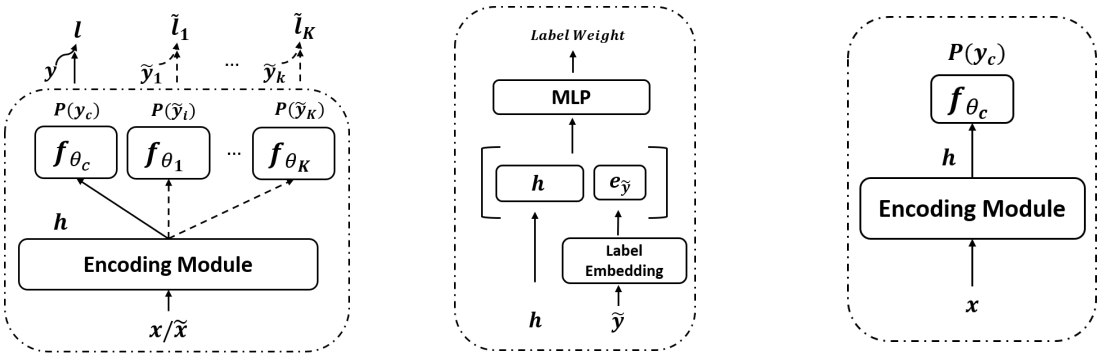
Problem Statement: Given a limited amount of manually annotated news data $\mathcal{D}$ and K sets of weakly labeled data $\{\tilde{\mathcal{D}}^{(k)}\}_{k=1}^{K}$ derived from K different weak labeling functions based on weak social signals, learn a fake news classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ which generalizes well onto unseen news pieces.

5.2 The Proposed Framework MWSS

User engagements over news articles, including posting about, commenting on or recommending the news, bear implicit judgments of the users about the news and could serve as weak sources of labels for fake news detection. For instance, prior research has shown that contrasting sentiment of users on a piece of news article, and similarly varying levels of credibility or bias, can be indicators of the underlying news being fake. However, these signals are noisy and need to be appropriately weighted for training supervised models. Due to the noisy nature of such social media engagements, we term these signals as *weak social supervision*.

To give a brief overview for the modeling, we define heuristic labeling functions (refer to Section 5.2.2) on user engagements to harvest such signals in order to weakly label a large amount of data. The weakly labeled data is combined with limited amount of manually annotated examples to build a fake news detection system that is better than training on either subset of the data. We emphasize that multiple weak labels can be generated for a single news article based on different labeling functions and we aim to jointly utilize both the clean examples as well as multiple sources of weak social supervision in this paper.

In this section, we focus on developing algorithms for the joint optimization of



(a) MWSS Classifier (b) MWSS LWN (c) MWSS inference

Figure 13: The proposed framework MWSS for learning with multiple weak supervision from social media data. (a) Classifier: Jointly modeling clean labels and weak labels from multiple sources; (b) LWN: Learning the label weight based on the concatenation of instance representation and weak label embedding vector. (c) During inference, MWSS uses the learned encoding module and classification MLP to predict labels for (unseen) instances in the test data.

manually annotated clean and multi-source weakly labeled instances in a unified framework.

5.2.1 Meta Label Weighting with Weak Social Supervision

Learning from multiple sources has shown promising performance in various domains such as truth discovery (Ge *et al.*, 2013), object detection (Ouyang *et al.*, 2014), etc. In this work, we have $K + 1$ distinct sources of supervision: clean labels coming from manual annotation and multiple sources of weak labels obtained from K heuristic labeling functions based on users' social engagements.

Our objective is to build an effective framework that leverages weak social supervision signals from multiple sources in addition to limited amount of clean data. However, signals from different weak sources are intrinsically noisy, biased in different

ways, and thus of varying degree of qualities. Simply treating all sources of weak supervision as equally important and merging them to construct a single large set of weakly supervised instances tend to result in sub-optimal performance–which we have used as a baseline in our experiments. However, it is challenging to determine the contribution of different sources of weak social supervision. To facilitate a principled solution of weighting weak instances, we leverage meta-learning. In this, we propose to treat label weighting as a meta-procedure, i.e., building a *label weighting network* (LWN) which takes an instance (e.g., news piece) and its weak label (obtained from social supervision) as input, and outputs a scalar value as the importance weight for the pair. The weight determines the contribution of the weak instance in training the desired fake news classifier in our context. The LWN can be learned by back-propagating the loss of the trained classifier on a separate clean set of instances. To allow information sharing among different weak sources, for the fake news classifier, we use a shared feature extractor to learn a common representation and use separate functions (specifically, MLPs) to map the features to different weak label sources.

Specifically, let $h_{\theta_E}(x)$ be an encoder that generates the content representation of an instance x with parameters θ_E. Note that this encoder is shared by instances from both the clean and multiple weakly labeled sources. Let $f_{\theta_c}(h(x))$ and $\{f_{\theta_k}(h(x))\}_{k=1,..,K}$ be the $K+1$ labeling functions that map the contextual representation of the instances to their labels on the clean and the K sets of weakly supervised data, respectively. In contrast to the shared parameters of the encoder θ_E, the parameters θ_c and $\{\theta_k\}_{k=1,...,K}$ are different for the clean and weak sources (learned by separate source-specific MLPs) to capture different mappings from the contextual representations to the labels from each source.

For training, we want to jointly optimize the loss functions defined over the (i) clean

while *not converged* **do**
  1. Update LWN parameters α by descending $\nabla_\alpha \mathcal{L}_{val}(\theta - \eta \nabla_\theta \mathcal{L}_{train}(\alpha, \theta))$
  2. Update classifier parameters θ by descending $\nabla_\theta \mathcal{L}_{train}(\alpha, \theta)$
end

Algorithm 1: Training process of MWSS

data and (ii) instances from the weak sources weighted by their respective utilities. The weight of the weak label $\tilde{y}$ for an instance x (encoded as $h(x)$) is determined by a separate Label Weighting Network (LWN) formulated as $\omega_\alpha(h(x), \tilde{y})$ with parameters α. Thus, for a given $\omega_\alpha(h(x), \tilde{y})$, the objective for training the predictive model with multiple sources of supervision jointly is:

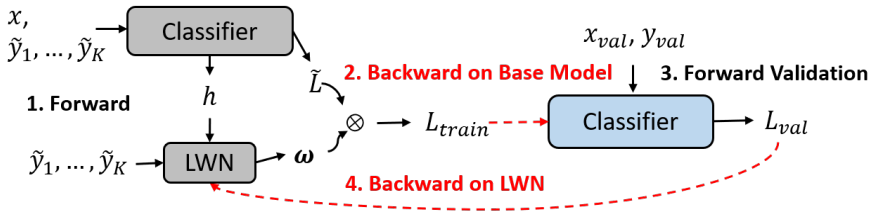
$$\min_{\theta_E, \theta_c, \theta_1, \ldots, \theta_k} \mathbb{E}_{(x,y)\in\mathcal{D}} \ell(y, f_{\theta_c}(h_{\theta_E}(x))) + \sum_{k=1}^{K} \mathbb{E}_{(x,\tilde{y})\in\tilde{\mathcal{D}}^{(k)}} \omega_\alpha(h_{\theta_E}(x), \tilde{y}) \ell(\tilde{y}, f_{\theta_k}(h_{\theta_E}(x))) \tag{5.1}$$

where ℓ denotes the loss function to minimize the prediction error of the model. The first component in the above equation optimizes the loss over the clean data, whereas the second component optimizes for the weighted loss (given by $w_\alpha(.)$) of the weak instances from K sources. Figure 13 shows the formulation for both the classifier and LWN.

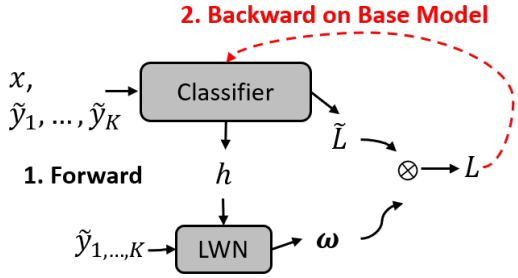
The final objective is to optimize LWN $\omega_\alpha(h(x), \tilde{y})$ such that when using such a weighting scheme to train the main classifier as specified by Eq. (5.1), the trained classifier can perform well on a separate set of clean examples. Formally, the following bi-level optimization problem describe the above intuition as:

$$\min_{\alpha} \mathcal{L}_{val}(\theta^*(\alpha)) \quad \text{s.t.} \quad \theta^* = \arg\min \mathcal{L}_{train}(\alpha, \theta) \tag{5.2}$$

where $\mathcal{L}_{train}$ is the objective in Eq. (5.1), θ denotes the concatenation of all classifier parameters $(\theta_E, \theta_c, \theta_1, ..., \theta_K)$, and $\mathcal{L}_{val}$ is loss of applying a trained model on a separate



(a) LWN Update



(b) Classifier Update

Figure 14: The illustration of the MWSS in two phases: (a) we compute the validation loss based on the validation dataset and retain the computation graph for LWN backward propagation; (b) the classifier updates its parameters through backward propagation on clean and weakly labeled data.

set of clean data. Note that $\theta^*(\alpha)$ denotes the dependency of θ^* on α after we train the classifier on a given LWN.

Analytically solving for the inner problem is typically infeasible. In this paper, we adopt the following one-step SGD update to approximate the optimal solution. As such the gradient for the meta-parameters α can be estimated as:

$$\nabla_\alpha \mathcal{L}_{val}(\theta - \eta \nabla_\theta \mathcal{L}_{train}(\alpha, \theta)) = -\eta \nabla^2_{\alpha,\theta} \mathcal{L}_{train}(\alpha, \theta) \nabla_{\theta'} \mathcal{L}_{val}(\theta') \tag{5.3}$$

$$\approx -\frac{\eta}{2\epsilon} \left[\nabla_\alpha \mathcal{L}_{train}(\alpha, \theta^+) - \nabla_\alpha \mathcal{L}_{train}(\alpha, \theta^-) \right] \tag{5.4}$$

where $\theta^\pm = \theta \pm \epsilon \nabla_{\theta'} \mathcal{L}_{val}(\theta')$, $\theta' = \theta - \eta \nabla_\theta \mathcal{L}_{train}(\alpha, \theta)$, ϵ is a small constant for finite difference and η is learning rate for SGD.

Since we leverage Multiple Weak Social Supervision, we term our method as MWSS.

We adopt Adam with mini-batch training to learn the parameters. Algorithm 4 and Figure 14 outline the training procedure for MWSS.

5.2.2 Constructing Weak Labels from Social Engagements

In this section, we describe how to generate weak labels from users' social engagements that can be incorporated as weak sources in our model.

Dataset Description: We utilize one of the most comprehensive fake news detection benchmark datasets called FakeNewsNet (Shu *et al.*, 2018b). The dataset is collected from two fact-checking websites: GossipCop[11] and PolitiFact[12] containing news contents with labels annotated by professional journalists and experts, along with social context information. News content includes meta attributes of the news (e.g., body text), whereas social context includes related users' social engagements on the news items (e.g., user comments in Twitter). Note that the number of news pieces in PolitiFact data is relatively small, and we enhance the dataset to obtain more weak labels. Specially, we use a news corpus spanning the time frame 01 January 2010 through 10 June 2019, from 13 news sources including mainstream British news outlets, such as BBC and Sky News, and English language versions of Russian news outlets such as RT and Sputnik, which are mostly related to political topics. To obtain the corresponding social engagements, we use a similar strategy as FakeNewsNet (Shu *et al.*, 2018b) to get tweets/comments, user profiles and user history tweets through the Twitter API and web crawling tools. For GossipCop data, we mask part of the

[11]https://www.gossipcop.com/

[12]https://www.politifact.com/

annotated data and treat them as unlabeled data for generating weak labels from the social engagements.

Generating Weak Labels: Now, we introduce the labeling functions for generating weak labels from social media via statistical measures guided by computational social theories.

First, research shows user opinions towards fake news have more diverse sentiment polarity and less likely to be neutral (Cui and Lee, 2019). So we measure the sentiment scores (using a widely used tool VADER (Hutto and Gilbert, 2014)) for all the users sharing a piece of news, and then measure the variance of the sentiment scores by computing the standard deviation. We define the following weak labeling function:

> ***Sentiment-based:*** *If a news piece has a standard deviation of user sentiment scores greater than a threshold τ_1, then the news is weakly labeled as fake news.*

Second, social studies have theorized the correlation between the bias of news publishers and the veracity of news pieces (Gentzkow *et al.*, 2015). Accordingly, we assume that news shared by users who are more biased are more likely be fake, and vice versa. Specifically, we adopt the method in (Kulshrestha *et al.*, 2017) to measure user bias (scores) by exploiting users' interests over her historical tweets. The hypothesis is that users who are more left-leaning or right-leaning share similar interests with each other. Following the method in (Kulshrestha *et al.*, 2017), we generate representative sets of people with known public bias, and then calculate bias scores based on how closely a query users' interests match with those representative users. We define the following weak labeling function:

> ***Bias-based:*** *If the mean value of users' absolute bias scores – sharing a piece of news – is greater than a threshold τ_2, then the news piece is weakly-labeled as fake news.*

Third, studies have shown that less credible users, such as malicious accounts or normal users who are vulnerable to fake news, are more likely to spread fake news (Shu *et al.*, 2017). To measure user credibility, we adopt the practical approach in (Abbasi and Liu, 2013). The hypothesis is that less credible users are more likely to coordinate with each other and form big clusters, whereas more credible users are likely to form small clusters. We use the hierarchical clustering[13] to cluster users based on their meta-information on social media and take the reciprocal of the cluster size as the credibility score. Accordingly, we define the following weak labeling function:

> ***Credibility-based:*** *If a news piece has an average credibility score less than a threshold τ_3, then the news is weakly-labeled as fake news.*

To determine the proper thresholds for τ_1, τ_2, and τ_3, we vary the threshold values from $[0, 1]$ through binary search, and compare the resultant weak labels with the true labels from the training set of annotated clean data – later used to train our meta-learning model – on GossipCop, and choose the value that achieves the the best accuracy on the training set. We set the thresholds as $\tau_1 = 0.15$, $\tau_2 = 0.5$, and $\tau_3 = 0.125$. Due to the sparsity for Politifact labels, for simplicity, we use the same threshold values as derived from the GossipCop data.

Quality of Weak Labeling Functions We apply the aforementioned labeling functions and obtain the weakly labeled positive instances. We treat the news pieces discarded by the weak labeling functions as *negative* instances. The statistics are shown in Table 8. To assess the quality of these weakly-labeled instances, we compare the weak labels with the true labels on the annotated clean data in GossipCop – later used to train our meta-learning model. Table 7 shows the confusion matrix in Gossip

[13]https://bit.ly/2WGK6zE

Table 7: Evaluation of weak labeling functions.

Weak Rule	Predictions	True Negative	True Positive
Sentiment	Negative	37%	63%
	Positive	16%	84%
Bias	Negative	67%	33%
	Positive	16%	84%
Credibility	Negative	67%	33%
	Positive	30%	70%

Table 8: The statistics of the datasets. Clean refers to manually annotated instances, whereas the weak ones are obtained by using the weak labeling functions

Dataset	GossipCop	Politifact
# Clean positive	1,546	303
# Clean negative	1,546	303
# Sentiment-weak positive	1,894	3,067
# Sentiment-weak negative	4,568	1,037
# Bias-weak positive	2,587	2,484
# Bias-weak negative	3,875	1,620
# Credibility-weak positive	2,765	2,963
# Credibility-weak negative	3,697	1,141

Cop dataset with tuned thresholds as discussed before. The accuracy of the weak labeling functions corresponding to Sentiment, Bias, and Credibility are 0.59, 0.74, 0.74, respectively. The F1-scores of these three weak labeling functions are 0.65, 0.64, 0.75. We observe that the accuracy of the labeling functions are significantly better than random (0.5) for binary classification indicating that the weak labeling functions are of acceptable quality.

5.3 Evaluating MWSS

In this section, we conduct experiments to evaluate the effectiveness of MWSS for early fake news detection. We aim to answer the following evaluation questions: (1) **EQ1**: Can MWSS improve fake news classification performance by leveraging weak social supervision; (2) **EQ2**: How effective are the different sources of supervision for improving prediction performance; and (3) **EQ3**: How robust is MWSS on leveraging multiple sources?

5.3.1 Experimental Settings

Evaluation measures. We use F1 score and accuracy as the evaluation metrics. We randomly choose 15% of the clean instances for validation and 10% for testing. We fix the number of weak training samples and select the amount of clean training data based on the *clean data ratio* defined as:

$$\text{clean ratio} = \frac{\#\text{clean labeled samples}}{\#\text{clean labeled samples} + \#\text{weak labeled samples}}.$$

This allows us to investigate the contribution of clean vs. weakly labeled data in later experiments. All the clean datasets are balanced with positive and negative instances. We report results on the test set with the model parameters picked with the best validation accuracy. All runs are repeated for 3 times and the average is reported.

Base Encoders. We use the convolutional neural networks (CNN) (Kim, 2014) and RoBERTa-base, a robustly optimized BERT pre-training model (Liu *et al.*, 2019) as the encoders for learning content representations. We truncate or pad the news text to 256 tokens, and for the CNN encoder we use pre-trained WordPiece embeddings from BERT to initialize the embedding layer. For each of the $K + 1$ classification heads,

we employ a two-layer MLP with 300 and 768 hidden units for both the CNN and RoBERTa encoders. The LWN contains a weak label embedding layer with dimension of 256, and a three-layer MLP with $(768, 768, 1)$ hidden units for each with a sigmoid as the final output function to produce a scalar weight between 0 and 1. We use binary cross-entropy as the loss function ℓ for MWSS[14].

Baselines and learning configurations. We consider the following settings:

(1) training only with limited amount of manually annotated **clean** data. Models include the following state-of-the-art early fake news detection methods:

- TCNN-URG (Qian *et al.*, 2018): This method exploits users' historical comments on news articles to learn to generate synthetic user comments. It uses a two-level CNN for prediction when user comments are not available for early detection.
- EANN (Wang *et al.*, 2018): This method utilizes an adversarial learning framework with an event-invariant discriminator and fake news detector. For a fair comparison, we only use the text CNN encoder.

(2) training only with **weakly** labeled data; and (3) training with both the **clean** and **weakly** labeled data as follows:

- Clean+Weak: In this setting, we simply merge both the clean and weak sets (essentially treating the weak labels to be as reliable as the clean ones) and use them together for training different encoders.
- L2R (Ren *et al.*, 2018): L2R is the state-of-the-art algorithm for learning to re-weight (L2R) examples for training models through a meta learning process.
- Snorkel (Ratner *et al.*, 2018b): It combines multiple labeling functions given their dependency structure by solving a matrix completion-style problem. We use

[14]All the data and code are available at: **this clickable link**

the label generated by Snokel as the weak label and feed it to the classification models.

- MWSS: The proposed model for jointly learning with clean data and multi-sources of weak supervision for early fake news detection.

Most of the above baseline models are geared for single sources. In order to extend them to multiple sources, we evaluated several aggregation approaches, and found that taking the majority label as the final label achieved the best performance result. We also evaluate an advanced multiple weak label aggregation method – Snorkel (Ratner *et al.*, 2019) as the multi-source baseline. Note that our MWSS model, by design, aggregates information from multiple sources and does not require a separate aggregation function like the majority voting.

5.3.2 Effectiveness of Weak Supervision and Joint Learning

To answer **EQ1**, we compare the proposed framework MWSS with the representative methods for fake news classification. We determine the model hyper-parameters with cross-validation. For example, we set parameters $learning\_rate \in \{10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}\}$ and choose the one that achieves the best performance on the held-out validation set. From Table 9, we make the following observations:

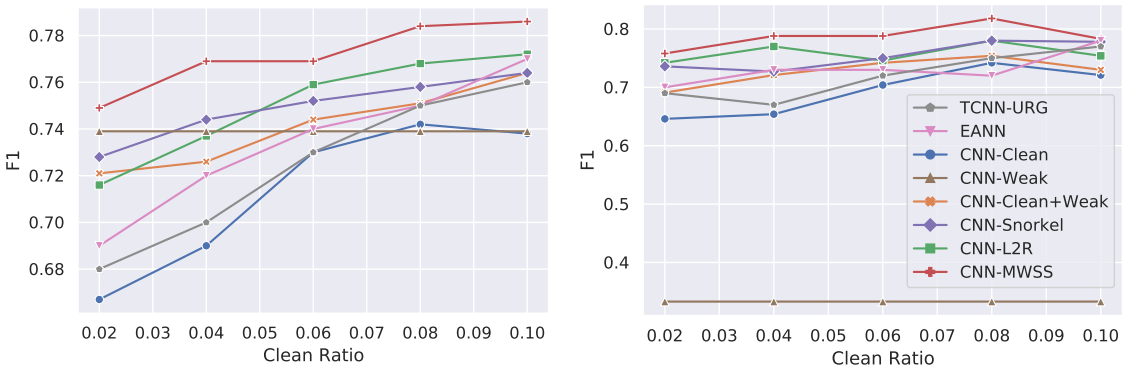
- Training only on clean data achieves better performance than training only on the weakly labeled data consistently across all the datasets (clean > weak).
- Among methods that only use clean data with CNN encoders, we observe TCNN-URG and EANN to achieve relatively better performance than CNN-clean consistently. This is because TCNN-URG utilizes user comments during training to capture additional information, while EANN considers the event

Table 9: Performance comparison for early fake news classification. *Clean* and *Weak* depict model performance leveraging only those subsets of the data; *Clean+Weak* is the union of both the sets.

Methods	GossipCop		PolitiFact	
	F1	Accuracy	F1	Accuracy
TCNN-URG (Clean)	0.76	0.74	0.77	0.78
EANN (Clean)	0.77	0.74	0.78	0.81
CNN (Clean)	0.74	0.73	0.72	0.72
CNN (Weak)	0.73	0.65	0.33	0.60
CNN (Clean+Weak)	0.76	0.74	0.73	0.72
CNN-*Snorkel* (Clean+Weak)	0.76	0.75	0.78	0.73
CNN-*L2R* (Clean+Weak)	0.77	0.74	0.79	0.78
CNN-MWSS (Clean+Weak)	**0.79**	**0.77**	**0.82**	**0.82**
RoBERTa (Clean)	0.77	0.76	0.78	0.77
RoBERTa (Weak)	0.74	0.74	0.33	0.60
RoBERTa (Clean+Weak)	0.80	0.79	0.73	0.73
RoBERTa-*Snorkel* (Clean+Weak)	0.76	0.74	0.78	0.77
RoBERTa-*L2R* (Clean+Weak)	0.78	0.75	0.81	0.82
RoBERTa-*MWSS* (Clean+Weak)	**0.80**	**0.80**	**0.82**	**0.82**

information in news contents (TCNN-URG>CNN-clean, and EANN>CNN-clean).

- On incorporating weakly labeled data in addition to the annotated clean data, the classification performance improves compared to that using only the clean labels (or only the weak labels) on both datasets (demonstrated by clean+weak, L2R, Snorkel > clean > weak).
- On comparing two different encoder modules, we find that RoBERTa achieves much better performance in GossipCop compared to CNN, and has a similar performance in PolitiFact. The smaller size of the PolitiFact data results in variable performance for RoBERTa.
- For methods that leverage both the weak and clean data, L2R and Snorkel



(a) GossipCop

(b) PolitiFact

Figure 15: F1 score with varying clean data ratio from 0.02 to 0.1 with CNN-MWSS. The trend is the similar with RoBERTa encoder (best visualized in color).

perform quite well. This is because L2R assigns weight to instances based on their contribution with a held-out validation set, whereas Snorkel leverages correlations across multi-source weak labeling functions to recover the label.

- In general, our model MWSS achieves the best performance. We observe that MWSS > L2R and Snorkel on both the datasets. This demonstrates the importance of treating weak labels differently from the clean labels with a joint encoder for learning shared representation, separate MLPs for learning source-specific mapping functions, and learning to re-weight instances via LWN. To understand the contribution of the above model components, we perform an ablation study in the following section.

5.3.3 Impact of the Ratio of Clean to Weakly Labeled Data on Classification Performance

To answer **EQ2**, we explore how the performance of MWSS changes with the clean ratio. We set the clean ratio to vary in $\{0.02, 0.04, 0.06, 0.08, 0.1\}$. To have

Table 10: F1/Accuracy on training MWSS on different weak sources with clean data.

Dataset	Sentiment	Bias	Credibility	All Sources
GossipCop	0.75/0.69	0.78/0.75	0.77/0.73	0.79/0.77
PolitiFact	0.75/0.75	0.77/0.77	0.75/0.73	0.78/0.75

a consistent setting, we fix the number of weakly labeled instances and change the number of clean labeled instances accordingly. In practise, we have abundant weak labels from the heuristic weak labeling functions. The objective here is to figure out how much clean labels to add in order to boost the overall model performance. Figure 15 shows the results. We make the following observations:

- With increasing values of clean ratio, the performance increases for all methods (except *Weak* which uses a fixed amount of weakly labeled data). This shows that increasing amount of reliable clean labels helps the models, as expected..
- For different clean ratio configurations, MWSS achieves the best performance compared to other baselines, i.e., MWSS > L2R and Snorkel. This shows that MWSS can more effectively utilize the clean and weak labels via its multi-source learning and re-weighting framework.
- We observe that the methods using Clean+Weak labels where we treat the weak labels to be as reliable as clean ones may not necessarily perform better than using only clean labels. This shows that simply merging the clean and weak sources of supervision without accounting for their reliability may not improve the prediction performance.

Table 11: F1/Accuracy result of ablation study on modeling source-specific MLPs with different clean ratio (C-Ratio). “SH" denotes a single shared MLP and “MH" denotes multiple source-specific ones.

Model	C-Ratio	L2R	LWN
SH	0.02	0.72/0.68	0.73/0.72
	0.10	0.77/0.74	0.77/0.73
MH	0.02	0.73/0.71	0.75/0.71
	0.10	0.78/0.76	0.79/0.77

5.3.4 Parameter Analysis

Impact of source-specific mapping functions:

In this experiment, we want to study the impact of modeling separate MLPs for source-specific mapping functions (modeled by f_{θ_k} in Equation 5.1) in LWN and L2R as opposed to replacing them with a single shared MLP (i.e. $f_{\theta_k} = f_\theta \ \forall k$) across multiple sources. From Table 11, we observe that MWSS and L2R both work better with multiple source-specific MLPs as opposed to a single shared MLP by better capturing source-specific mapping functions from instances to corresponding weak labels. We also observe that MWSS performs better than L2R for the respective MLP configurations – demonstrating the effectiveness of our re-weighting module.

Impact of different weak sources: To study the impact of multi-source supervision, we train MWSS separately with individual weak sources of data along with clean annotated instances with a clean ratio of 0.1. From Table 10, we observe that training MWSS with multiple weak sources achieves better performance compared to that of a single weak source – indicating complementary information from different weak sources help the model. To test whether MWSS can capture the quality of each source,

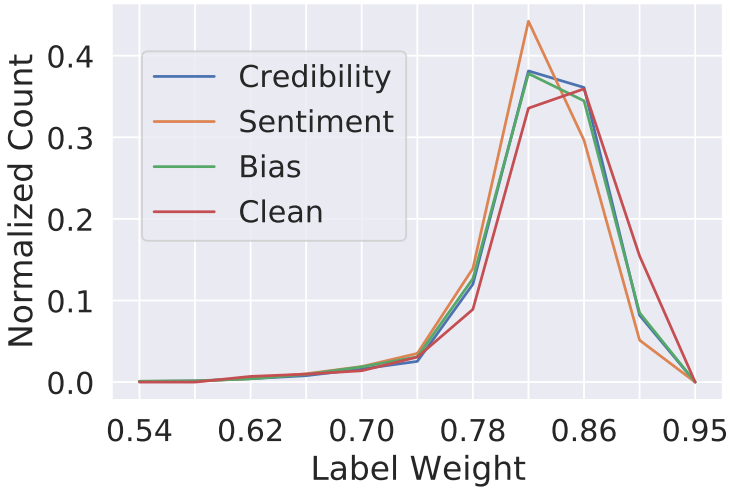


Figure 16: Label weight density distribution among weak and clean instances in GossipCop. The mean of the label weights for weak sources from *Credibility-based*, *Sentiment-based*, *Bias-based*, and *Clean* are 0.86, 0.85, 0.86 and 0.87 respectively.

we visualize the label weight distribution for each weak source and clean dataset in Figure 16.

From the weight distribution, we also observe the weight of the sentiment-source (referred as *Sentiment*) to than that of other sources. In addition, although the LWN is not directly trained on clean samples, it still assigns the largest weight to the clean source. These demonstrate that our model not only learns the importance of different instances but also learns the importance of the corresponding source.

Chapter 6

CROSS-DOMAIN FAKE NEWS DETECTION

In this chapter, we study cross-domain fake news detection. The content of fake news is shown to be rather diverse in terms of topics, styles and media platforms (Shu *et al.*, 2017). For a real-world fake news detection system, it is often unrealistic to obtain abundant labeled data for every domain (e.g., Entertainments and Politics are two different domains) due to the expensive labeling cost. As such, fake news detection is commonly performed in the single-domain setting, and supervised (Wang *et al.*, 2018) or unsupervised methods (Hosseinimotlagh and Papalexakis, 2018; Yang *et al.*, 2019) are proposed to handle limited or even unlabeled domains. However, the performance is largely limited due to overfitting on small labeled samples or without any supervision information. In addition, models learned on one domain, known as the source domain, may be biased to event-specific features. Hence, it is natural and necessary to explore auxiliary information to improve fake news prediction with limited labels for other domains, called target domains.

One way to tackle this problem is to utilize domain adaptation techniques to explore the auxiliary information to transfer the knowledge from the source domain to the target domain. Domain adaptation has shown success for addressing domain discrepancies in many tasks including text and image classification (Tzeng *et al.*, 2017), sentiment analysis (Blitzer *et al.*, 2007), and recommendations (Fernández-Tobías *et al.*, 2012). In addition, studies have shown that fake news publishers often have intent to spread distorted and misleading information and influence large communities of consumers, requiring particular writing styles necessary to appeal to and persuade

a wide scope of consumers that is not seen in true news (Shu *et al.*, 2017; Afroz *et al.*, 2012). Therefore, it has great potential to capture the domain-independent features to predict fake news across domains. Based on domain adaptation, we propose to learn transferable knowledge from one news domain to another that can be more robust and general for fake news detection across domains.

In addition to transferring knowledge across domains, we can also exploit auxiliary information within a single domain from rich social context information. Social context refers to the social environment where news disseminates, which naturally provides additional information to supplement and enhance news content-based detection methods. Users can express their opinions and sentiments by posting/commenting on the news pieces. For example, comments like *"This didn't happen because president does not have the power to give citizenship."*, help us determine whether a news piece on *"Obama secretly gave citizenship to 2,500 people"* is fake or not. Thus, we can jointly exploit the complementary information of news contents and user comments for fake news detection. Moreover, the users can participate in propagating the news by tweeting about it or re-tweet the news. This process leads to user-news interactions. Leveraging the user-news interaction helps us to further improve the fake news detection systems as users have consistent behaviors in sharing fake or real news.

Cross-domain knowledge transfer and within-domain joint learning offer complementary information, providing new perspectives to help improve the prediction performances for fake news in a newly emerging domain. Therefore, we would like to study the novel and challenging problem of cross-domain fake news detection. In essence, we solve the following challenges: (1) how to perform cross-domain fake news detection effectively; (2) how to learn domain adaptive representations across domains;

and (3) how to capture news contents, user comments, and user-news interactions to detect fake news. Our solutions to above challenges result in a novel framework named as CrossFND (Cross-domain Fake News Detection). Next, I will introduce the details for the proposed framework CrossFND.

6.1 Problem Statement

Let $\mathcal{D}_s = \{(\mathbf{x}_1^s, y_1^s), (\mathbf{x}_2^s, y_2^s), ..., (\mathbf{x}_N^s, y_N^s)\}$ and $\mathcal{D}_t = \{(\mathbf{x}_1^t, y_1^t), (\mathbf{x}_2^t, y_2^t), ..., (\mathbf{x}_M^t, y_M^t)\}$ denote a set of N news pieces and their labels and M news pieces and labels from source and target domain, respectively. Each news may have a set of comments $\mathbf{c}_i \in \mathcal{C}$ and user-news interactions $\mathbf{u}_i \in \mathcal{U}$ which provide extra information about the news. The user-news interaction $\mathbf{u}_i$ is a binary vector indicating the users who have tweeted or re-tweeted about the news $\mathbf{x}_i$. Each news $\mathbf{x}_i$ consists of a sequence of words $\{w_1, w_2, ..., w_K\}$. The goal of the domain adaptive fake news classifier is to learn a cross domain representation for news $\mathbf{x}_i$ such that it can classify fake news from different domains. In this paper, we study the following problem:

Problem Statement: Given two sets of news pieces $\mathcal{D}_s$ and $\mathcal{D}_t$, corresponding user comments $\mathcal{C}_s$ and $\mathcal{C}_t$, and users interactions $\mathcal{U}_s$ and $\mathcal{U}_t$ from the source and target domains respectively, learn a classifier F that can classify fake news in the target domain accurately.

6.2 The Proposed Framework CrossFND

In this section, we discuss the domain adaptive fake news classifier. The input of our model is the source set $\mathcal{D}_s$ and a portion γ of target set $\mathcal{D}_t$. As it is shown in Figure 17, our model CrossFND consists of several components: (1) news information

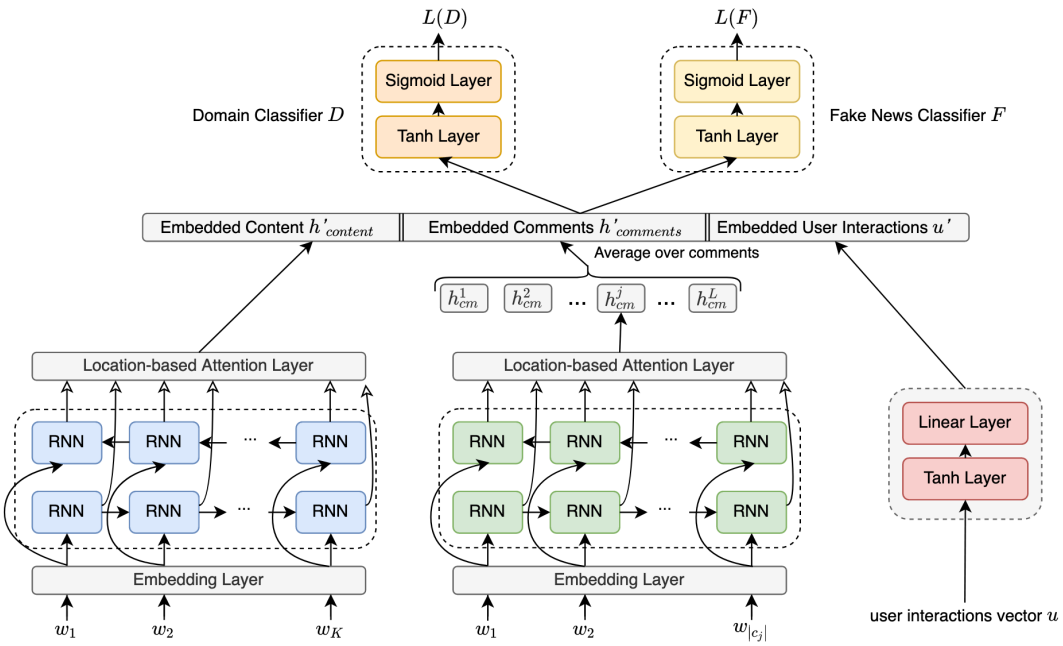


Figure 17: The architecture of the proposed model CrossFND.

encoders, (2) a fake news classifier, and, (3) a domain classifier. First, we go through the domain adaptive representation learning in CrossFND, then, we discuss news content, comments, and user-news interactions fusion.

6.2.1 Domain-Adaptive Representation Learning

We first introduce the proposed domain-adaptive representation learning. The problem of classifying fake news requires to learn representations of the word sequence in the news content. RNNs are particularly good at modeling sequential data. In practice, both Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are capable of modeling sequential data. For our study, we specifically use bi-directional GRU. One of the most important characteristics of bidirectional neural networks,

is that they are able to understand context more efficiently when compared to the traditional unidirectional networks (Kiperwasser and Goldberg, 2016).

To create domain adaptive representations, each news $\mathbf{x} = \{w_1, w_2, ..., w_K\}$ is passed to an embedding layer to create a representation for each word w_k. The embedding layer stores a tensor of dimension $|V| \times d$, in which V is the vocabulary set and d is the dimension of embedded words. Initially, these tensors will be assigned according to GloVe word embeddings (Pennington *et al.*, 2014). This layer maps discrete words into compact vectors. Each vector in the sequence corresponds to a word from the news content. To further capture the contextual information of annotations, we use bidirectional GRU to model word sequences from both directions of words. The bidirectional GRU contains the forward GRU $\overrightarrow{f}$ which reads news x from word w_1 to w_k and a backward GRU $\overleftarrow{f}$ which reads news x from word w_K to w_1:

$$\begin{aligned} \overrightarrow{\mathbf{h}_i} &= \overrightarrow{GRU}(\mathbf{w}_i), i \in \{1, \ldots, K\} \\ \overleftarrow{\mathbf{h}_t^i} &= \overleftarrow{GRU}(\mathbf{w}_i), i \in \{K, \ldots, 1\} \end{aligned} \tag{6.1}$$

We then obtain an annotation of word w_i by concatenating the forward hidden state $\overrightarrow{\mathbf{h}_i}$ and backward hidden state $\overleftarrow{\mathbf{h}_i}$, i.e., $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_i}]$, which contains the information of the whole sentence centered around w_i.

However, one notable drawback of encoders, is that they cannot easily handle long sentences. Since, the news sentences are usually long, it may be difficult for the neural network to manage long sentences, and hence it quickly gets intractable (Chung *et al.*, 2014).To address this problem, we use fixed length sentences and introduce an attention mechanism. After the encoder generates the final output and hidden states, they are fed to the attention layer.

Attention layer helps the classifier to generate a more comprehensive context vector.

To this end, we use location-based attention based on (Luong *et al.*, 2015) which calculates an attention value a_i for each output $\mathbf{h}_i$ showing the importance of that value. Having the attention vector $\mathbf{a}$, the resultant context vector can be calculated as $\mathbf{h}' = \sum_i a_i \mathbf{h}_i$.

After extracting the context vectors of news contents, we train a classifier to classify fake news. The problem by using only a classifier to training and testing it on a dataset is that it can overfit on the given dataset and only extracts the discriminative features among the data which are from the source domain. It is obvious that in different domains, different features are important to achieve good classification performance. In our experiment, we will show that if we simply train a fake news classifier in one domain, it is not guaranteed that it would work in another domain as well. In this paper, our goal is to build a cross-domain classifier. Hence, we introduce another component, domain classifier, to help our model to find differences between the source and target domains.

The domain classifier is responsible for detecting the domain of a news content. Thus, we use a portion of target dataset to train the classifier. By adding the domain classifier component to our model, it would push the encoder to learn representations which are domain independent. The domain classifier would correctly classify whether an instance is from the source or target domain if it has domain related features. During the optimization part of our method (Section 6.2.3), we try to train our model to fool this classifier, resulting in a domain independent textual representation.

6.2.2 News Content, Comments, and User-News Interactions Fusion

In most of social media, each news content has different comments. Comments can help us to decide whether a news is fake or real (Shu *et al.*, 2018b). Comments like *"This is so not true"* gives us a hint that the news may be fake. To leverage the use of comments, we add the comments encoder. This component creates a context vector for each comment using a similar method to news content encoder. We calculate the average of the context of comments to use in our model.

While considering news content and its comments may give us an idea on how fake it might be, it is more practical to check the users who have shared the news as well. To incorporate users metadata in our model, we consider using user-news interactions. User-news interactions $\mathbf{u}$ for news x is a binary vector where $\mathbf{u}_j = 1$ indicates that the user j has tweeted or re-tweeted about news $\mathbf{x}$. To encode this information into our model, we consider using a feed forward network. This network gets the $\mathbf{u}$ as input and after passing it through the hidden and linear layers, it gives us an encoded vector containing information about users interactions with news $\mathbf{x}$. As in Figure 17, after encoding the comments and user-news interactions, we concatenate the vectors before passing them to the classifier components.s

6.2.3 An Optimization Algorithm

Now we introduce each component and the corresponding training process: (1) a news content, comments, and user-news interactions encoder; (2) a fake news classifier; and (3) a domain classifier.

Representation Encoding: This part extracts the representations of news content,

comments, and user-news interactions. Each news content $\mathbf{x}_i$ is passed through an embedding layer which replaces each word w with a vector. Then, each word vector is passed through a bi-directional GRU to calculate final hidden state $(\overrightarrow{\mathbf{h}_N}, \overleftarrow{\mathbf{h}_N})$ and outputs $\{(\overrightarrow{\mathbf{h}_1}, \overleftarrow{\mathbf{h}_1}), (\overrightarrow{\mathbf{h}_2}, \overleftarrow{\mathbf{h}_2}), ..., (\overrightarrow{\mathbf{h}_N}, \overleftarrow{\mathbf{h}_N})\}$.

After this step, each forward and backward hidden state are concatenated as $\mathbf{h}_t = (\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t})$ and are passed through an attention layer to calculate the context vector. The same procedure goes for the comments. Moreover, the user-news interactions' binary vector is passed through a feed forward neural network to compute its embedding:

$$\mathbf{u}' = \tanh(\mathbf{W}_u^{(1)}\mathbf{u} + \mathbf{b}_u^{(1)})\mathbf{W}_u^{(2)} + \mathbf{b}_u^{(2)} \tag{6.2}$$

where **(W, b)** are learnable weights.

Finally, the three representation vectors of news content, comments, and user-news interactions are combined to create the final context vector $\mathbf{h}'$. Note that for news content without comments, we concatenate a zero vector to the context vector of news content.

Fake news classifier: This component is trained based on whether an input context vector is fake news or not. An input vector is passed through a three layer neural network classifier to calculate the output as follows:

$$\mathbf{o} = \tanh(\mathbf{W}_F^{(1)}\mathbf{h}' + \mathbf{b}_F^{(1)}) \tag{6.3}$$

$$p_F = \text{sigmoid}(\mathbf{W}_F^{(2)}\mathbf{o} + \mathbf{b}_F^{(2)}) \tag{6.4}$$

where $\mathbf{W}_F^{(1)}$ ($\mathbf{W}_F^{(2)}$), $\mathbf{b}_F^{(1)}$ ($\mathbf{b}_F^{(2)}$) and p_F are network weights, bias and output probability for news x respectively.

Domain classifier: This component is trained based on whether an input context is from the source or target domain. This classifier is a three layer neural network

similar to fake news classifier, but with different weights and bias matrix $(\mathbf{W}_D, \mathbf{b}_D)$. This classifier outputs probability p_D.

To combine these components, we use the following loss function to calculate the loss value for the fake news classifier F and the domain classifier D:

$$L = -\frac{1}{M}\sum_{i=1}^{M}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)) \tag{6.5}$$

Finally, we train the model to optimize the following loss function:

$$\min_{F}\max_{D} \alpha L(F) - \beta L(D) \tag{6.6}$$

We want our model to learn domain independent features. To this end, we try to maximize the domain loss $L(D)$ for the model to learn domain independent features. This will force the model to create a representation for news contents in a way that does not show the domain specific features, hence tries to fool the domain classifier. On the other hand, we want our model to classify fake news with good performance. So we minimize the fake news classifier's loss function $L(F)$. To control the balance between loss values, we consider two parameters α and β showing the contribution of fake news and domain classifier.

6.3 Evaluating CrossFND

In this section, we conduct experiments to evaluate the effectiveness of our algorithm and answer the following questions: **Q1**: How well our method can detect fake news on its *source domain*?, **Q2**: While trained on a source domain A, how well our method can detect fake news on a *target domain* B?, and, **Q3**: How effective are news contents and user comments in improving the detection performance of CrossFND?

Platform	**G**	**P**
# True News	3,586	2,645
# Fake News	2,230	2,770
# News	5,819	5,415
# News with Comments	5,819	415
# Users	43,918	60,053
# Unique Users	100,520	

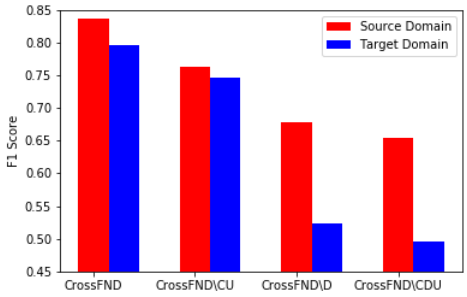
Table 12: The statistics of the datasets

To answer the first question (**Q1**), we train CrossFND on the training set and report the results for the testing set in the source domain. For the second question (**Q2**), we train CrossFND on a source domain A and use target domain B to test. Finally, we study the impact of news content, comments and user-news interactions for classifying fake news to answer the last question (**Q3**).

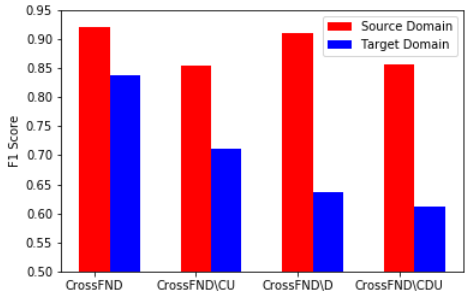
6.3.1 Experimental Settings

Datasets: We utilize the benchmark data repository FakeNewsNet (Shu *et al.*, 2018b). The FakeNewsNet includes data from two well-known fact-checking platforms *GossipCop* (G for short) and *Politifact* (P for short). GossipCop consists of news about the gossip industry, while Politifact includes political news. This dataset has two labels {*Fake, Real*}. Due to the inadequate data from *Politifact*, we enrich this dataset with another Politifact dataset from (Rashkin *et al.*, 2017). The new dataset consists of $16,000$ news pieces from which we uniformly select $5,000$ news to concatenate with the FakeNewsNet's Politifact dataset. Table 12 shows the final statistics of dataset.

Implementation Details: In this part we go through the parameters and implementation details of our model. Before we give each news content to the model, we



(a) **PolitiFact as Source Domain**



(b) **GossipCop as Source Domain**

Figure 18: Impact analysis of news contents, comments, and user-news interactions for fake news detection.

select the first 512 words from each news content as our input data. The same method goes for the comments. The intuition behind this is that each news is very long and training the model on a long text is not efficient.

Each news content and comments will be passed through a bi-directional GRU to capture the initial context. The bi-directional GRU has 1 hidden layer with size of 128. As for the comments encoder, we use a smaller hidden size of 64.

Finally, we pass each initial context and the bi-directional GRU outputs through the attention layer to get the final context vector. Note that if a news content does not have comments or user-news interactions, the context for these parts would be a zero vector. The final context vector is then passed to both news content and domain classifier. The news content classifier is a three layer neural network in which the hidden layer size is 512 and the output size is 1. Moreover, we use a hidden layer consisting of 512 neurons and an output layer with 128 neurons to create user-news interactions' embedding. Note that the embedding layer is initialized with GloVe (Pennington *et al.*, 2014) weights.

Experimental Design: We use different baselines for comparison:

- **TCNN-URG** (Qian *et al.*, 2018) Similar to the TCNN model (Kim, 2014), this model incorporates convolutional neural network to model news contents, and can capture different granularity of text features. This model uses an additional component to use user responses for classification.
- **CSI** (Ruchansky *et al.*, 2017); CSI is a hybrid deep learning model that utilizes information from text, response, and source. The news representation is modeled via an LSTM neural network with the Doc2Vec (Le and Mikolov, 2014) embedding on the news contents and user comments as input, and for a fair comparison, the user features are ignored.
- **dEFEND** (Shu *et al.*, 2019a) dEFEND develops a sentence-comment co-attention network to exploit both news contents and user comments to capture explainable sentences and user comments for fake news detection.
- **BERT-BiLSTM** (Vlad *et al.*, 2019) This model uses a large network including BERT (Vaswani *et al.*, 2017a), a bidirectional LSTM layer, and a capsule layer to classify text using transfer learning technique.

Note that for fair comparison, we consider state-of-the-art baselines that: (1) only use news contents including BERT-BiLSTM, (2) use both news contents and user comments including TCNN-URG and dEFEND, and (3) considers users interactions including the CSI method. In addition, the proposed CrossFND and BERT-BiLSTM can handle cross-domain predictions.

6.3.2 Performance on Cross-domain Fake News Detection

To evaluate our method and answer questions (**Q1**) and (**Q2**), we train our model on a source domain A, then we report the AUC and F1 for both source domain A

	CrossFND	CSI	TCNN-URG	dEFEND	BERT-BiLSTM
G → P	0.932 (0.921)	0.832 (0.811)	0.792 (0.770)	0.893 (0.857)	0.947 (0.918)
P → G	0.853 (0.837)	0.785 (0.756)	0.741 (0.526)	0.793 (0.768)	0.868 (0.824)

Table 13: Source domain results. The values show AUC (F1) on the testing set from source domain.

	CrossFND	CSI	TCNN-URG	dEFEND	BERT-BiLSTM
G → P	**0.887** (**0.871**)	0.581 (0.547)	0.450 (0.448)	0.712 (0.634)	0.748 (0.711)
P → G	**0.835** (**0.795**)	0.612 (0.598)	0.545 (0.471)	0.583 (0.452)	0.634 (0.598)

Table 14: Target domain results. The values show AUC (F1) on the target domain.

and target domain B. Note that in these experiments we use 10-fold validation on source data. In our further experiments we study the impact of comments, user-news interactions, and parameters. For the sake of fairness, we use combination of source domain, and 30% of target domain data to train baseline models as well. The reason we chose 30% is that according to Figure 19c, the performance difference after using more than 30% of target domain data is not significant.

Source Domain (Q1). Table 13 shows the performance of CrossFND on the source domain. $A \rightarrow B$ means we use A as the source domain and B as the target domain. According to this table, after training, the model can detect fake news from source domain very well. Comparing to other methods, CrossFND has around 25% improvement over TCNN-URG. While BERT-BiLSTM achieves a comparable performance, it takes much time and resource for training. Finally, CrossFND also manages to achieve a better performance than dEFEND.

Target Domain (Q2). Table 14 shows the performance of CrossFND on the target domain. This experiment is the same as above, but we report the results on the target domain. As illustrated, our model can detect fake news from target domain

	Source Domain	Target Domain
G → P	0.921 (0.864)	0.710 (0.654)
P → G	0.743 (0.724)	0.531 (0.432)

Table 15: Results of CrossFND without using domain adaptation. The numbers indicate AUC (F1 score).

with a high AUC and F1 score while other methods overfit on the source domain and cannot detect fake news from target domain very well.

Combining these two results, we can ensure that our model successfully works on both the source and target domains. In the following experiments we study the impact of each component in our proposed method. Furthermore, other methods *CSI*, *TCNN-URG*, and *dEFEND* can achieve an acceptable performance on the source domain, but they fail to detect fake news from another domain. This may be a result of overfitting on the source domain.

News Content, Comments and User-News Interactions Fusion (Q3). To show the impact of different components, we define different variants of CrossFND:

- CrossFND\CU we remove both comments and user-news interactions encoder.
- CrossFND\D: we remove the domain classifier.
- CrossFND\CD: we remove both comments encoder and domain classifier.

Table 15 shows the results on CrossFND\D without using the domain classifier as an adversary. The results indicate that the two domains are different from each other and training on one domain would not guarantee a good performance on the other one.

By using the user-news interactions and partial comments we have an performance improvement in both domains. In addition, we also try to detect fake news without

using domain adaptation but including the comments and user-news interactions. The results illustrate that the best case is to use the domain adaptation component and the extra information to achieve the best performance.

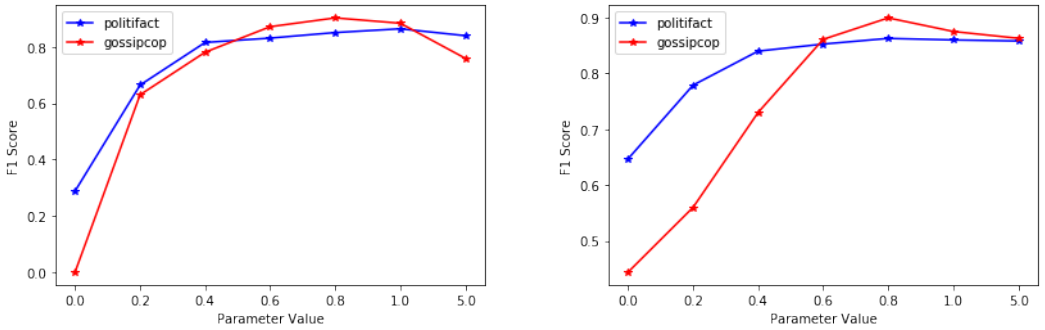
Figure 18 shows the results on different variants of our model, which can confirm using both domain adaptation and comments will improve the results. Furthermore, during investigating the comments, we can see that most of them do not have any information about the news content. Some comments are: *the title is misleading*, and *Its fake He doesn't have a daughter*. Although most of the comments do not have information about the news content specifically, they carry enough information for classifying fake news in different domains.

6.3.3 Parameter Analysis

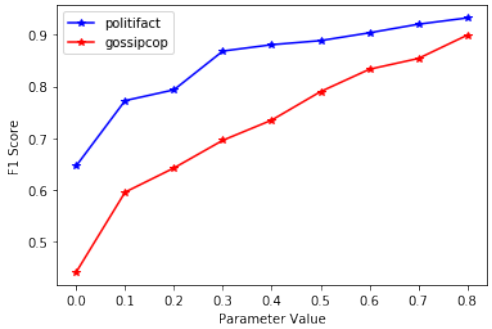
Our proposed method has two parameters α and β for changing the effect of $L(F)$ and $L(D)$, respectively. We illustrate the effect of these parameters by changing it as $\alpha, \beta \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 5.0\}$. Furthermore, we also illustrate how much portion of target data we need to get an acceptable performance on both domains. We use a portion γ of target domain data in range of $\gamma \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$.

In Figure 19a, we can see the F1 score for different values of α. The α parameters controls the amount of contribution for fake news classifier's loss function. In this experiment we use a constant value of $\beta = 0.8$ while changing the α value. From this result, we can consider using $\alpha = 1.0$ is a good choice for fake news classifier's loss contribution.

To analyze the impact of β, we use value of $\alpha = 1.0$ and different values of β to see how β impacts the performance of CrossFND. Figure 19b shows the F1 score



(a) **Impact of** α (b) **Impact of** β



(c) Impact of γ

Figure 19: Impact of different parameters on CrossFND's performance on target domain. Blue line shows the result of trained model on *gossipcop* while the red one shows the F1 for model trained on *politifact*.

on the target domain for different values of β. By using $\beta = 0.0$, the model does not use domain component for fake news detection, hence its performance on the target domain would be low. By increasing β, CrossFND uses a higher loss value from domain classifier, hence pushes the encoder to learn a domain independent textual representation. From the results, as there is no huge performance difference after $\beta = 0.5$, we consider this value for other experiments.

Finally, we study the impact of target domain data portion γ for training our model. In this test, we train our model on one domain and report the results for the other domain. The results are shown in Figure 19c. In this figure, blue line shows the testing results for using Politifact data as the target domain, and the red line shows

the results for using Gossipcop as the target domain. While using more target domain data leads to better performance, it is not a good practice to use all of the target domain data. The reason is that training a model on both domains need abundant data and takes time. Furthermore, in some cases there is not enough data from a specific domain, hence, it is important that a domain adaptive model works on partial data from the target domain. In our case, from Figure 19c, we can infer that using 30% of target domain data is enough for getting an acceptable performance. From the parameter analysis results, we can conclude that our model is stable for different values of α and β. Using similar values would not result in huge performance decrease or increase.

Chapter 7

CONCLUSION AND FUTURE WORK

In this chapter, we summarize our research results and their broader impacts, and discuss promising research directions.

7.1 Summary

In this dissertation, I investigate weak social supervision learning for understanding disinformation. I provide principled approaches of exploiting social theories to design algorithms to tackle these challenges of disinformation and fake news detection in challenging scenarios. In particular, I study four innovative research tasks - (1) effective fake news detection; (2) explainable fake news detection; (3) early fake news detection; and (4) cross-domain fake news detection. I propose novel frameworks to tackle these challenges not only from textual content but also the unique properties of related social media engagements to detect fake news.

For effective fake news detection, we propose a novel framework TriFN to model tri-relationship among publishers, news pieces, and consumers, for fake news detection. TriFN can extract effective features from news publisher and user engagements separately, as well as capture the interrelationship simultaneously. We make several important findings of disinformation spreading from large-scale social media data: (1) publishers with a high degree of bias are more likely to publish disinformation; (2) consumers with low credibilities are more likely to spread disinformation, while users with high credibility scores are less likely to spread disinformation. These findings

serve as the groundwork of a semi-supervised framework TriFN for effective fake news detection.

For explainable fake news detection, we formally define the problem and propose a principled approach called dEFEND to learn feature representations for fake news detection and explainable sentences/comments discovery. We observe that the user comments that are related to the content of original news pieces are helpful to detect fake news and explain prediction results. The successful experiences of conducting explainable fake news detection suggest that: (1) disinformation can be mixed with truth and false, and some information is more check-worthy; and (2) consumers' feedback to disinformation can be informative and helpful to explain disinformation. Experiments on real-world datasets demonstrate the effectiveness of the proposed framework.

For early fake news detection, we investigate the early detection of fake news with multi-source weak social supervision. We proposed principled ways to construct multi-source weak labels from user engagements with weak labeling rules. We find that: (1) weak labels derived from social supervision have complementary information to the content to detect disinformation; (2) different sources of weak supervision can have different impacts on the disinformation detection performance. Experiments on real-world datasets demonstrate the effectiveness of our framework MWSS for detecting disinformation at an early stage.

For cross-domain fake news detection, we proposed a principled approach CrossFND to predict fake news in newly emerging domains when limited data is available. CrossFND utilizes domain adaptation techniques to explore the auxiliary information to transfer the knowledge from the source domain to the target domain. We find that: (1) domain-independent feature representations are helpful to identify disinformation

besides textual signals; and (2) user profiles and their comments that are related to the content provide auxiliary signals to detect disinformation. Experiments on real-world datasets demonstrate that CrossFND can outperform baselines for detecting fake news in both the source domain and target domain.

Overall, this dissertation investigates learning with weak social supervision for understanding disinformation with the following computational tasks: effective fake news detection, explainable fake news detection, early fake news detection, and cross-domain fake news detection. These four novel algorithms facilitate the exploration on other types of disinformation. The methodologies and techniques presented in this dissertation fall into a novel paradigm, i.e., learning with weak social supervision, that has important implications in broadening applications in social media.

7.2 Future Work

Learning with weak social supervision for understanding disinformation is still in its early stages of development and an active area of exploration. Below we present some promising research directions:

- **Beyond Disinformation Detection: Attribution, Characterization, and Mitigation** Detecting disinformation is an essential step to build computational tools to identify disinformation on a large-scale. However, to better combat disinformation and reduce its detrimental effects, we need to look beyond disinformation detection.
  - For ***attribution***, the goal is to verify the purported source or provider and the associated attribution evidence. Attribution search in social media is a new problem because social media lacks a centralized authority or

mechanism that can store and certify provenance of a piece of social media data. From a network diffusion perspective, to identify the provenance is to find a set of key nodes such that the information propagation is maximized (Shu *et al.*, 2018a). Identifying provenance paths can indirectly find the originated provenances. The provenance paths of information are usually unknown, and for disinformation and misinformation in social media it is still an open problem. The provenance paths delineate how information propagates from the sources to other nodes along the way, including those responsible for retransmitting information through intermediaries. One can utilize the characteristics of social media to trace back to the source (Gundecha *et al.*, 2013). Based on the Degree Propensity and Closeness Propensity hypotheses (Barbier *et al.*, 2013), the nodes with higher degree centralities that are closer to the nodes are more likely to be transmitters. Hence, it is estimated that the top transmitters from the given set of potential provenance nodes are obtained through graph optimization. We plan to develop new algorithms which can incorporate information other than the network structure such as the node attributes and temporal information to better discover provenances.

With the success of deep learning especially deep generative models, machine-generated text can be a new type of fake news that is fluent, readable, and catchy, which brings about new attribution sources. For example, benefiting from the adversarial training, a series of language generation models are proposed such as SeqGAN (Yu *et al.*, 2017), MaliGAN (Che *et al.*, 2017), LeakGAN (Guo *et al.*, 2018b), MaskGAN (Fedus *et al.*, 2018), etc. and unsupervised models based on Transformer (Vaswani

et al., 2017b) using multi-task learning are proposed for language generation such as GPT-2 (Radford *et al.*, 2019) and Grover (Zellers *et al.*, 2019). One important problem is to consider machine-generated synthetic text and propose solutions to differentiate which models are used to generate these text. One can perform classification on different text generation algorithms' data and explore the decision boundaries. The collections of data can be acquired from representative language generation models such as VAE, SeqGAN, TextGAN, MaliGAN, GPT-2, Grover, etc.

– For ***characterization***, the goal is to understand whether the information is malicious, has harmless intents, or has other insightful traits. When people create and distribute disinformation they typically have a specific purpose in mind, or intent. For example, there can be many possible intents behind the deception including: (1) persuade people to support individuals, groups, ideas, or future actions; (2) persuade people to oppose individuals, groups, ideas or future actions; (3) produce emotional reactions (fear, anger or joy) toward some individual, group, idea or future action in the hope of promoting support or opposition; (4) educate (e. g., about vaccination threat); (5) prevent an embarrassing or criminal act from being believed; (6) exaggerate the seriousness of something said or done (e.g., use of personal email by government officials); (7) create confusion over past incidents and activities (e. g., did the U.S. really land on the moon or just in a desert on earth?); or (8) demonstrate the importance of detecting disinformation to social platforms (e. g., Elizabeth Warren and Mark Zuckerberg dispute). End to end models augmented with feature embeddings such as causal relations between claims and evidence can be

used (Hidey and McKeown, 2016) to detect the intents such as Persuasive influence detection (Hidey and McKeown, 2018). Once we have identified the intent behind a deceptive news article, we can further understand how successful this intent will be: what is the likelihood that this intent will be successful in achieving its intended purpose. We can consider measures of virality grounded in social theories to aid characterization. Social psychology points to social influence (how widely the news article has been spread) and self-influence (what preexisting knowledge a user has) as viable proxies for drivers of disinformation dissemination (Zhou *et al.*, 2019b). Greater influence from the society and oneself skews a user's perception and behavior to trust a news article and to unintentionally engage in its dissemination. Computational social network analysis (Shu *et al.*, 2018a) can be used to study how social influence affects behaviors and/or beliefs of individuals exposed to disinformation and fake news.

– For ***mitigation***, the goal is to proactively block target users or start a mitigating campaign at an early stage. To achieve this, key spreaders of fake news need to be discovered such as provenances and persuaders . In addition, estimating the potential population affected by a fake news is useful for decision-makers to mitigate otherwise influential fake news. Moreover, choosing specific users to block the cascade of fake news, and even to start mitigation campaigns to immunize users are required to minimize the influence of fake news. First, identifying key users on social media is important to mitigate the effect of fake news. For example, the provenances of fake news indicates the sources or originators. Provenances can help answer questions such as whether the piece of news has been modified during

its propagation, and how an "owner" of the piece of information is connected to the transmission of the statement. In addition, it's necessary to identify influential persuaders to limit the spread scope of fake news by blocking the information flow from them to their followers on social media. Second, the fake news diffusion process has different stages in terms of people's attention and reactions over time, resulting in a unique life cycle different from that of in-depth news (Castillo *et al.*, 2014). The impact of fake news on social media can be estimated as the number of users that are potentially affected by the news piece, an amount we want to assess and then minimize. We can adapt the network scale-up based method (Bernard *et al.*, 1990) for estimating the size of uncountable populations to estimate the size of the population affected by fake news on the provenance paths discussed previously. Third, network intervention is to develop strategies to control the widespread dissemination of fake news before it goes viral. Network intervention mainly consists of two perspectives: influence Minimization and Mitigation Campaign. On the one hand, limiting the spread of fake news can be seen as analogous to inoculation in the face of an epidemic. Models of epidemics generally assume that a global parameter describes the probability that a user is infected by a neighbor. This assumption is violated in real-world situations of information exchange where users have varying degrees of willingness to accept information from their neighbors. Thus, the Independent Cascade Model (ICM) (Saito *et al.*, 2008) is proposed to alleviate this problem by assuming each edge has its specific activation probability. On the other hand, limiting the impact of fake news is not only to minimize the spread of fake news but also maximize the spread of

true news. The campaign to mitigate fake news and to maximize true news forms during the information diffusion process. The network activities of fake news and real news can be represented as Multivariate Hawks Processes (MHP) with self and mutual excitations, where the control incentives more spontaneous mitigation events (Farajtabar *et al.*, 2017).

- **Neural Disinformation Generation and Detection** Fake news has been an important problem on social media and is amplified by the powerful deep learning models due to their power of generating neural fake news (Zellers *et al.*, 2019). In terms of neural fake news generation, recent progress allows malicious users to generate fake news based on limited information. Models like Generative Adversarial Network (GAN) (Guo *et al.*, 2018b) can generate long readable text from noise and GPT-2 (Radford *et al.*, 2019) can write news stories and fiction books with simple context. Existing fake news generation approaches may not be able to produce style-enhanced and fact-enriched text, which preserves the emotional/catchy styles and relevant topics related to news claims. Detecting these neural fake news pieces firstly requires us to understand the characteristic of these news pieces and detection difficulty. Dirk Hovy *et al.* propose an adversarial setting in detecting the generated reviews (Hovy, 2016). (Zellers *et al.*, 2019) and (Solaiman *et al.*, 2019) propose neural generation detectors that fine-tune classifiers on generator's previous checkpoint. It is important and interesting to explore: i) how to generate realistic fake news with neural generative models to better understand neural-generated fake news? ii) can we differentiate human-generated and machine-generated fake/real news?
- **Techniques for Learning Weak Social Supervision** We expect along the direction of learning with weak social supervision, more research will emerge in

the near future. First, leveraging weak social supervision for computation social science research is promising. Since computational social science research usually relies on relatively limited offline survey data, weak social supervision can serve as a powerful online resource to understand and study social computing problems. Second, existing approaches utilize single or combine multiple sources of weak social supervision, while to what extent and aspect the weak social supervision helps is fairly important to explore. Third, the capacity of ground-truth labels and weak social supervision and the relative importance between the sources are essentials to develop learning methodology in practical scenarios. Moreover, the weak supervision rules may have complementary information since they capture social signals from different perspectives. An interesting future direction is to explore multi sources of weak social supervision in a principled way to model the mutual benefits through data programming.

- **AI for Good** AI for Good holds a lot of potentials as well as challenges for solving societal problems beyond combating the disinformation. AI, or machine intelligence, is when machines are programmed with some (but not all) aspects of human intelligence, including learning, problem solving and prioritization. While social goods mean services that benefit the large number of people. The intersection of AI and social goods is AI for social goods that aims to close the gap between human and machine to power services that benefit a large number of people. To empower AI for good, I discuss three research directions relevant to the theme for this dissertation: (1) building explanation-enabled AI systems; (2) AI with limited data; and (3) illuminating AI with interdisciplinary research.
  - ***Building Explanation-enabled AI Systems*** The unparalleled computing power combined with big data has shown unprecedented success in many

applications containing image, graphs, text and speech. However, most of the current methodologies are data-driven and conducted in a passive fashion, and thus are inadequate to apprehend knowledge from human intents and demands. Thus, it is important to go beyond the state-of-the-art machine learning methods, adapting the learning strategies and acquiring knowledge from human feedback. We hope the research can fundamentally change the decision-making process with valuable explanations in various domains (e.g., social media, e-commerce, education, and security) and improve the utilities of existing data-driven AI algorithms. To establish the fundamental principles for building the explanation-enabled AI systems, it is important to explore : (1) how to transform the meaningful human cognition into knowledge? (2) how to incorporate noisy, incomplete, and complicated human feedback for better representation learning? (3) how to interpret prediction results with knowledge reasoning and causal discovery.

– ***AI with limited data*** The majority of this dissertation aims to tackle the variety property of big data. However, real-world applications such as social media consistently and continuously generates massive content at an unprecedented rate. On the other hand, social media data often has limited labels. Therefore, it suggests the necessity to provide solutions to the big data when facing the problem of limited labels. In practice, we may have access to small amount of annotated training examples and large amount of weakly labeled examples constructed from weak social supervision. I have introduced how to jointly learn with strong and weak social supervision to detect fake news at an early stage, which is based on my previous work on learning with weak supervision for email intent detection. With these

preliminary and on-going work, It is to investigate principled approaches for inventing effective AI systems with limited data and to cooperate with industry to leverage the value from big data to directly help people and society.

– ***Illuminating AI with Interdisciplinary Research*** Real-world data such as social media data is big and complex. It opens the door to interdisciplinary research and allows researchers to collectively study large-scale human behavior otherwise impossible. Powered by knowledge, data is the new oil for AI. However, data-driven approaches may face challenges. In many real-world applications, labeled data is often limited due to the expensive annotation cost or privacy of data access. Social media data provides unique characteristics that are suitable to derive user patterns guided by interdisciplinary disciplines such as social theories, cognitive theories, neural science, etc. Learning from interdisciplinary disciplines brings a new paradigm to advance AI. This dissertation benefits from various sociology and journalism studies to guide the learning with social supervision. An another example, brain network analysis plays a vital role in understanding some biologically fundamental mechanisms of the human brain. My previous work collaborating with a biologist is to study the connections of brain areas with diseases using AI algorithms (Zhang *et al.*, 2018). There are many potentials to close the gap of AI with interdisciplinary research in real-world applications such as education, health care, agriculture, medicine, energy, etc.

REFERENCES

Abbasi, M. A. and H. Liu, "Measuring user credibility in social media.", in "SBP", pp. 441–448 (Springer, 2013).

Afroz, S., M. Brennan and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online", in "ISSP", (2012).

Bahdanau, D., P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville and Y. Bengio, "An actor-critic algorithm for sequence prediction", arXiv preprint arXiv:1607.07086 (2016).

Barbier, G., Z. Feng, P. Gundecha and H. Liu, "Provenance data in social media", Synthesis Lectures on Data Mining and Knowledge Discovery **4**, 1, 1–84 (2013).

Bernard, H. R., E. C. Johnsen, P. D. Killworth, C. McCarty, G. A. Shelley and S. Robinson, "Comparing four different methods for measuring personal social networks", Social networks **12**, 3, 179–215 (1990).

Blitzer, J., M. Dredze and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification", in "Proceedings of the 45th annual meeting of the association of computational linguistics", pp. 440–447 (2007).

Boyd, S. and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

Castillo, C., M. El-Haddad, J. Pfeffer and M. Stempeck, "Characterizing the life cycle of online news stories using social media reactions", in "Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing", pp. 211–223 (ACM, 2014).

Castillo, C., M. Mendoza and B. Poblete, "Information credibility on twitter", in "Proceedings of the 20th international conference on World wide web", pp. 675–684 (ACM, 2011).

Che, T., Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Song and Y. Bengio, "Maximum-likelihood augmented discrete generative adversarial networks", arXiv preprint arXiv:1702.07983 (2017).

Chen, T. and C. Guestrin, "Xgboost: A scalable tree boosting system", in "KDD", (2016).

Chung, J., C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", arXiv preprint arXiv:1412.3555 (2014).

Ciampaglia, G. L., P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer and A. Flammini, "Computational fact checking from knowledge networks", PloS one **10**, 6, e0128193 (2015).

Cui, L. and S. W. D. Lee, "Same: Sentiment-aware multi-modal embedding for detecting fake news", (2019).

Davis, C. A., O. Varol, E. Ferrara, A. Flammini and F. Menczer, "Botornot: A system to evaluate social bots", in "Proceedings of the 25th International Conference Companion on World Wide Web", pp. 273–274 (International World Wide Web Conferences Steering Committee, 2016).

Farajtabar, M., J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song and H. Zha, "Fake news mitigation via point process based intervention", arXiv preprint arXiv:1703.07823 (2017).

Fedus, W., I. Goodfellow and A. M. Dai, "Maskgan: better text generation via filling in the_", arXiv preprint arXiv:1801.07736 (2018).

Feng, S., R. Banerjee and Y. Choi, "Syntactic stylometry for deception detection", in "ACL", pp. 171–175 (Association for Computational Linguistics, 2012).

Fernández-Tobías, I., I. Cantador, M. Kaminskas and F. Ricci, "Cross-domain recommender systems: A survey of the state of the art", in "Spanish conference on information retrieval", p. 24 (sn, 2012).

Frénay, B. and M. Verleysen, "Classification in the presence of label noise: a survey", IEEE transactions on neural networks and learning systems **25**, 5, 845–869 (2013).

Ge, L., J. Gao, X. Li and A. Zhang, "Multi-source deep learning for information trustworthiness estimation", in "Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 766–774 (ACM, 2013).

Gentzkow, M., J. M. Shapiro and D. F. Stone, "Media bias in the marketplace: Theory", Tech. rep., National Bureau of Economic Research (2014).

Gentzkow, M., J. M. Shapiro and D. F. Stone, "Media bias in the marketplace: Theory", in "Handbook of media economics", vol. 1, pp. 623–645 (Elsevier, 2015).

Guacho, G. B., S. Abdali, N. Shah and E. E. Papalexakis, "Semi-supervised content-based detection of misinformation via tensor embeddings", in "2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)", pp. 322–325 (IEEE, 2018).

Gundecha, P., Z. Feng and H. Liu, "Seeking provenance of information using social media", in "Proceedings of the 22nd ACM international conference on Information & Knowledge Management", pp. 1691–1696 (ACM, 2013).

Guo, H., J. Cao, Y. Zhang, J. Guo and J. Li, "Rumor detection with hierarchical social attention network", in "Proceedings of the 27th ACM International Conference on Information and Knowledge Management", pp. 943–951 (ACM, 2018a).

Guo, J., S. Lu, H. Cai, W. Zhang, Y. Yu and J. Wang, "Long text generation via adversarial training with leaked information", in "Thirty-Second AAAI Conference on Artificial Intelligence", (2018b).

Gupta, A., H. Lamba, P. Kumaraguru and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy", in "WWW", (2013).

Hassan, N., F. Arslan, C. Li and M. Tremayne, "Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster", in "Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 1803–1812 (ACM, 2017).

Hendrycks, D., M. Mazeika, D. Wilson and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise", in "Advances in neural information processing systems", pp. 10456–10465 (2018).

Hidey, C. and K. McKeown, "Identifying causal relations using parallel wikipedia articles", in "Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 1424–1433 (2016).

Hidey, C. T. and K. McKeown, "Persuasive influence detection: The role of argument sequencing", in "Thirty-Second AAAI Conference on Artificial Intelligence", (2018).

Hosseinimotlagh, S. and E. E. Papalexakis, "Unsupervised content-based identification of fake news articles with tensor decomposition ensembles", (2018).

Hovy, D., "The enemy in your own camp: How well can we detect statistically-generated fake reviews–an adversarial study", in "Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)", pp. 351–356 (2016).

Hutto, C. J. and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text", in "Eighth international AAAI conference on weblogs and social media", (2014).

Ireton, C. and J. Posetti, "Journalism,'fake news'& disinformation", (2018).

Järvelin, K. and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques", TOIS (2002).

Ji, Y. and J. Eisenstein, "Representation learning for text-level discourse parsing", in "ACL'2014", vol. 1, pp. 13–24 (2014).

Jin, Z., J. Cao, Y.-G. Jiang and Y. Zhang, "News credibility evaluation on microblog with a hierarchical propagation model", in "ICDM", pp. 230–239 (IEEE, 2014).

Jin, Z., J. Cao, Y. Zhang and J. Luo, "News verification by exploiting conflicting social viewpoints in microblogs.", in "AAAI", pp. 2972–2978 (2016).

Karimi, H., P. Roy, S. Saba-Sadiya and J. Tang, "Multi-source multi-class fake news detection", in "Proceedings of the 27th International Conference on Computational Linguistics", pp. 1546–1557 (2018).

Kim, Y., "Convolutional neural networks for sentence classification", arXiv preprint arXiv:1408.5882 (2014).

Kingma, D. P. and M. Welling, "Auto-encoding variational bayes", arXiv preprint arXiv:1312.6114 (2013).

Kiperwasser, E. and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional lstm feature representations", Transactions of the Association for Computational Linguistics **4**, 313–327 (2016).

Kulshrestha, J., M. Eslami, J. Messias, M. B. Zafar, S. Ghosh, K. P. Gummadi and K. Karahalios, "Quantifying search bias: Investigating sources of bias for political searches in social media", in "CSCW", (2017).

Kumar, S., R. West and J. Leskovec, "Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes", in "Proceedings of the 25th international conference on World Wide Web", pp. 591–602 (2016).

Kwon, S., M. Cha, K. Jung, W. Chen and Y. Wang, "Prominent features of rumor propagation in online social media", in "ICDM'13", pp. 1103–1108 (IEEE, 2013).

Le, Q. and T. Mikolov, "Distributed representations of sentences and documents", in "International Conference on Machine Learning", pp. 1188–1196 (2014).

Lee, D. D. and H. S. Seung, "Algorithms for non-negative matrix factorization", in "Advances in neural information processing systems", pp. 556–562 (2001).

Li, Y., J. Yang, Y. Song, L. Cao, J. Luo and L.-J. Li, "Learning from noisy labels with distillation", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 1910–1918 (2017).

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach", arXiv preprint arXiv:1907.11692 (2019).

Lu, J., J. Yang, D. Batra and D. Parikh, "Hierarchical question-image co-attention for visual question answering", in "Advances in neural information processing systems", pp. 289–297 (2016).

Luong, M.-T., H. Pham and C. D. Manning, "Effective approaches to attention-based neural machine translation", arXiv preprint arXiv:1508.04025 (2015).

Ma, J., W. Gao, Z. Wei, Y. Lu and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites", in "CIKM", (2015).

Magdy, A. and N. Wanas, "Web-based statistical fact checking of textual documents", in "Proceedings of the 2nd international workshop on Search and mining user-generated contents", pp. 103–110 (ACM, 2010).

Mele, N., D. Lazer, M. Baum, N. Grinberg, L. Friedland, K. Joseph, W. Hobbs and C. Mattsson, "Combating fake news: An agenda for research and action", (2017).

Meng, Y., J. Shen, C. Zhang and J. Han, "Weakly-supervised hierarchical text classification", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 33, pp. 6826–6833 (2019).

Mercier, H., "How gullible are we? a review of the evidence from psychology and social science", Review of General Psychology **21**, 2, 103–122 (2017).

Mitra, T. and E. Gilbert, "Credbank: A large-scale social media corpus with associated credibility annotations.", in "ICWSM", (2015).

Murphy, D., "Fake news 101", Independently published (2019).

Nettleton, D. F., A. Orriols-Puig and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques", Artificial intelligence review **33**, 4, 275–306 (2010).

Ouyang, W., X. Chu and X. Wang, "Multi-source deep learning for human pose estimation", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 2329–2336 (2014).

Pan, R. and M. Scholz, "Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering", in "KDD", (2009).

Patrini, G., A. Rozza, A. Krishna Menon, R. Nock and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 1944–1952 (2017).

Pauca, V. P., F. Shahnaz, M. W. Berry and R. J. Plemmons, "Text mining using non-negative matrix factorizations", in "SDM", (2004).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python", JMLR **12**, Oct, 2825–2830 (2011).

Pennebaker, J. W., R. L. Boyd, K. Jordan and K. Blackburn, "The development and psychometric properties of liwc2015", Tech. rep. (2015).

Pennington, J., 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)", pp. 1532–1543 (2014).

Potthast, M., J. Kiesel, K. Reinartz, J. Bevendorff and B. Stein, "A stylometric inquiry into hyperpartisan and fake news", arXiv preprint arXiv:1702.05638 (2017).

Qian, F., C. Gong, K. Sharma and Y. Liu, "Neural user response generator: Fake news detection with collective user intelligence.", in "IJCAI", (2018).

Quattrociocchi, W., A. Scala and C. R. Sunstein, "Echo chambers on facebook", (2016).

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language models are unsupervised multitask learners", OpenAI Blog **1**, 8 (2019).

Rashkin, H., E. Choi, J. Y. Jang, S. Volkova and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking", in "Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing", pp. 2931–2937 (2017).

Ratner, A., S. H. Bach, H. Ehrenberg, J. Fries, S. Wu and C. Ré, "Snorkel: Rapid training data creation with weak supervision", Proceedings of the VLDB Endowment **11**, 3, 269–282 (2017).

Ratner, A., S. H. Bach, H. Ehrenberg, J. Fries, S. Wu and C. Ré, "Snorkel: Rapid training data creation with weak supervision", The VLDB Journal pp. 1–22 (2019).

Ratner, A., B. Hancock, J. Dunnmon, F. Sala, S. Pandey and C. Ré, "Training complex models with multi-task weak supervision", arXiv preprint arXiv:1810.02840 (2018a).

Ratner, A., B. Hancock, J. Dunnmon, F. Sala, S. Pandey and C. Ré, "Training complex models with multi-task weak supervision", (2018b).

Reed, S., H. Lee, D. Anguelov, C. Szegedy, D. Erhan and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping", arXiv preprint arXiv:1412.6596 (2014).

Ren, M., W. Zeng, B. Yang and R. Urtasun, "Learning to reweight examples for robust deep learning", arXiv preprint arXiv:1803.09050 (2018).

Riedel, B., I. Augenstein, G. P. Spithourakis and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task", arXiv preprint arXiv:1707.03264 (2017).

Rubin, V. L., N. Conroy and Y. Chen, "Towards news verification: Deception detection methods for news discourse", in "Hawaii International Conference on System Sciences", (2015).

Rubin, V. L. and T. Lukoianova, "Truth and deception at the rhetorical structure level", Journal of the Association for Information Science and Technology **66**, 5, 905–917 (2015).

Ruchansky, N., S. Seo and Y. Liu, "Csi: A hybrid deep model for fake news", arXiv preprint arXiv:1703.06959 (2017).

Saito, K., R. Nakano and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model", in "International Conference on Knowledge-Based and Intelligent Information and Engineering Systems", pp. 67–75 (Springer, 2008).

Santia, G. C. and J. R. Williams, "Buzzface: A news veracity dataset with facebook user commentary and egos.", in "ICWSM", (2018).

Shahnaz, F., M. W. Berry, V. P. Pauca and R. J. Plemmons, "Document clustering using nonnegative matrix factorization", Information Processing & Management **42**, 2, 373–386 (2006).

Shang, J., J. Shen, T. Sun, X. Liu, A. Gruenheid, F. Korn, Á. D. Lelkes, C. Yu and J. Han, "Investigating rumor news using agreement-aware search", in "Proceedings of the 27th ACM International Conference on Information and Knowledge Management", pp. 2117–2125 (ACM, 2018).

Shao, C., G. L. Ciampaglia, A. Flammini and F. Menczer, "Hoaxy: A platform for tracking online misinformation", in "WWW", (2016).

Sharma, K., F. Qian, H. Jiang, N. Ruchansky, M. Zhang and Y. Liu, "Combating fake news: A survey on identification and mitigation techniques", arXiv preprint arXiv:1901.06437 (2019).

Shu, K., A. H. Awadallah, S. Dumais and H. Liu, "Detecting fake news with weak social supervision", IEEE Intelligent Systems (2020a).

Shu, K., H. R. Bernard and H. Liu, "Studying fake news via network analysis: Detection and mitigation", CoRR **abs/1804.10233** (2018a).

Shu, K., 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", pp. 395–405 (2019a).

Shu, K., D. Mahudeswaran and H. Liu, "Fakenewstracker: a tool for fake news collection, detection, and visualization", Computational and Mathematical Organization Theory pp. 1–12 (2019b).

Shu, K., D. Mahudeswaran, S. Wang, D. Lee and H. Liu, "Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media", arXiv preprint arXiv:1809.01286 (2018b).

Shu, K., S. Mukherjee, G. Zheng, A. H. Awadallah, M. Shokouhi and S. Dumais, "Learning with weak supervision for email intent detection", SIGIR (2020b).

Shu, K., A. Sliva, S. Wang, J. Tang and H. Liu, "Fake news detection on social media: A data mining perspective", ACM SIGKDD Explorations Newsletter **19**, 1, 22–36 (2017).

Shu, K., S. Wang, D. Lee and H. Liu, "Mining disinformation and fake news: Concepts, methods, and recent advancements", arXiv preprint arXiv:2001.00623 (2020c).

Shu, K., S. Wang and H. Liu, "Understanding user profiles on social media for fake news detection", in "2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)", pp. 430–435 (IEEE, 2018c).

Shu, K., S. Wang and H. Liu, "Beyond news contents: The role of social context for fake news detection", in "Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining", pp. 312–320 (ACM, 2019c).

Solaiman, I., M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford and J. Wang, "Release strategies and the social impacts of language models", arXiv preprint arXiv:1908.09203 (2019).

Stewart, R. and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge", in "AAAI", (2017).

Sukhbaatar, S., J. Bruna, M. Paluri, L. Bourdev and R. Fergus, "Training convolutional networks with noisy labels", arXiv preprint arXiv:1406.2080 (2014).

Tacchini, E., G. Ballarin, M. L. Della Vedova, S. Moret and L. de Alfaro, "Some like it hoax: Automated fake news detection in social networks", arXiv preprint arXiv:1704.07506 (2017).

Tzeng, E., J. Hoffman, K. Saenko and T. Darrell, "Adversarial discriminative domain adaptation", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 7167–7176 (2017).

Varma, P., F. Sala, A. He, A. Ratner and C. Ré, "Learning dependency structures for weak supervision models", ICML (2019).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need", in "Advances in neural information processing systems", pp. 5998–6008 (2017a).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need", (2017b).

Vlad, G.-A., M.-A. Tanase, C. Onose and D.-C. Cercel, "Sentence-level propaganda detection in news articles with transfer learning and bert-bilstm-capsule model", in "Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda", pp. 148–154 (2019).

Vosoughi, S., D. Roy and S. Aral, "The spread of true and false news online", Science **359**, 6380, 1146–1151 (2018).

Wang, W. Y., ""' liar, liar pants on fire"': A new benchmark dataset for fake news detection", arXiv preprint arXiv:1705.00648 (2017).

Wang, Y., 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", pp. 849–857 (ACM, 2018).

Wu, L. and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate", (2018).

Wu, L., F. Morstatter, K. M. Carley and H. Liu, "Misinformation in social media: definition, manipulation, and detection", ACM SIGKDD Explorations Newsletter **21**, 2, 80–90 (2019).

Wu, Y., P. K. Agarwal, C. Li, J. Yang and C. Yu, "Toward computational fact-checking", VLDB **7**, 7, 589–600 (2014).

Xu, W., X. Liu and Y. Gong, "Document clustering based on non-negative matrix factorization", in "Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval", pp. 267–273 (ACM, 2003).

Yang, S., K. Shu, S. Wang, R. Gu, F. Wu and H. Liu, "Unsupervised fake news detection on social media: A generative approach", in "AAAI", (2019).

Yu, L., W. Zhang, J. Wang and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient", in "Thirty-First AAAI Conference on Artificial Intelligence", (2017).

Zafarani, R., M. A. Abbasi and H. Liu, *Social media mining: an introduction* (Cambridge University Press, 2014).

Zellers, R., A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner and Y. Choi, "Defending against neural fake news", arXiv preprint arXiv:1905.12616 (2019).

Zhang, W., K. Shu, S. Wang, H. Liu and Y. Wang, "Multimodal fusion of brain networks with longitudinal couplings", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 3–11 (Springer, 2018).

Zhang, X. and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion", Information Processing & Management (2019).

Zheng, G., A. H. Awadallah and S. Dumais, "Meta label correction for learning with weak supervision", arXiv preprint arXiv:1911.03809 (2019).

Zhou, X., J. Cao, Z. Jin, F. Xie, Y. Su, D. Chu, X. Cao and J. Zhang, "Real-time news cer tification system on sina weibo", in "Proceedings of the 24th International Conference on World Wide Web", pp. 983–988 (ACM, 2015).

Zhou, X. and R. Zafarani, "Fake news: A survey of research, detection methods, and opportunities", arXiv preprint arXiv:1812.00315 (2018).

Zhou, X., R. Zafarani, K. Shu and H. Liu, "Fake news: Fundamental theories, detection strategies and challenges", in "Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining", pp. 836–837 (ACM, 2019a).

Zhou, X., R. Zafarani, K. Shu and H. Liu, "Fake news: Fundamental theories, detection strategies and challenges", in "WSDM", (2019b).

Zhou, Z.-H., "A brief introduction to weakly supervised learning", National Science Review **5**, 1, 44–53 (2018).

APPENDIX A

AN OPTIMIZATION OF TRIFN

In this chapter, we present the detail optimization process for the proposed framework TriFN. The objective for TriFN is as follows.

$$\begin{aligned}\min_{\mathbf{D},\mathbf{U},\mathbf{V},\mathbf{T}\geq 0,\mathbf{p},\mathbf{q}} & \|\mathbf{X}-\mathbf{D}\mathbf{V}^T\|_F^2+\alpha\|\mathbf{Y}\odot(\mathbf{A}-\mathbf{U}\mathbf{T}\mathbf{U}^T)\|_F^2 \\ & +\beta\mathrm{tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})+\gamma\|\mathbf{e}\odot(\bar{\mathbf{B}}\mathbf{D}\mathbf{q}-\mathbf{o})\|_2^2 \\ & +\eta\|\mathbf{D}_L\mathbf{p}-\mathbf{y}_L\|_2^2+\lambda R\end{aligned} \tag{A.1}$$

If we update the variables jointly, the objective function in Eq. A.1 is not convex. Thus, we propose to use alternating least squares to update the variables separately. For simplicity, we user $\mathcal{L}$ to denote the objective function in Eq. A.1. Next, we introduce the updating rules for each variable in details.

Update $\mathbf{D}$. Let Ψ_D be the Lagrange multiplier for constraint $\mathbf{D}\geq 0$, the Lagrange function related to $\mathbf{D}$ is,

$$\begin{aligned}\min_{\mathbf{D}} & \|\mathbf{X}-\mathbf{D}\mathbf{V}^T\|_F^2+\beta\mathrm{tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})+\gamma\|\mathbf{e}\odot(\bar{\mathbf{B}}\mathbf{D}\mathbf{q}-\mathbf{o})\|_2^2 \\ & +\eta\|\mathbf{D}_L\mathbf{p}-\mathbf{y}_L\|_2^2+\lambda\|\mathbf{D}\|_F^2-\mathrm{tr}(\Psi_D\mathbf{D}^T)\end{aligned} \tag{A.2}$$

and $\mathbf{D}=[\mathbf{D}_L;\mathbf{D}_U]$ and $\mathbf{H}=[\mathbf{U};\mathbf{D}_L]$. We rewrite $\mathbf{L}=[\mathbf{L}_{11},\mathbf{L}_{12};\mathbf{L}_{21},\mathbf{L}_{22}]$, where $\mathbf{L}_{11}\in\mathbb{R}^{m\times m}$, $\mathbf{L}_{12}\in\mathbb{R}^{m\times r}$,$\mathbf{L}_{21}\in\mathbb{R}^{r\times m}$, and $\mathbf{L}_{22}\in\mathbb{R}^{r\times r}$; and $\mathbf{X}=[\mathbf{X}_L,\mathbf{X}_U]$. The partial derivative of $\mathcal{L}$ w.r.t. $\mathbf{D}$ as follows,

$$\begin{aligned}\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\mathbf{D}} & =(\mathbf{D}\mathbf{V}^T-\mathbf{X})\mathbf{V}+\lambda\mathbf{D}+\gamma\bar{\mathbf{B}}^T\mathbf{E}^T(\mathbf{E}\bar{\mathbf{B}}\mathbf{D}\mathbf{q}-\mathbf{E}\mathbf{o})\mathbf{q}^T \\ & +\big[\beta\mathbf{L}_{21}\mathbf{U}+\beta\mathbf{L}_{22}\mathbf{D}_L+\eta(\mathbf{D}_L\mathbf{p}-\mathbf{y}_L)\mathbf{p}^T;\mathbf{0}\big]-\Psi_D\end{aligned} \tag{A.3}$$

where $\mathbf{E}\in\mathbb{R}^{l\times l}$ is a diagonal matrix with $\{\mathbf{e}_k\}_{k=1}^l$ on the diagonal and zeros everywhere else. By setting the derivative to zero and using Karush-KuhnTucker complementary condition (Boyd and Vandenberghe, 2004), i.e., $\Psi_D(i,j)\mathbf{D}_{ij}=0$,we get,

$$\mathbf{D}_{ij}\leftarrow\mathbf{D}_{ij}\sqrt{\frac{\hat{\mathbf{D}}(i,j)}{\tilde{\mathbf{D}}(i,j)}} \tag{A.4}$$

$$\begin{aligned}\hat{\mathbf{D}} & =\mathbf{X}\mathbf{V}+\gamma\big(\bar{\mathbf{B}}^T\mathbf{E}^T\mathbf{E}\mathbf{o}\mathbf{q}^T\big)^+ +\gamma\big(\bar{\mathbf{B}}^T\mathbf{E}^T\mathbf{E}\bar{\mathbf{B}}\mathbf{D}\mathbf{q}\mathbf{q}^T\big)^- \\ & +\big[\eta\big(\mathbf{D}_L\mathbf{p}\mathbf{p}^T\big)^-+\eta\big(\mathbf{y}_L\mathbf{p}^T\big)^+ +\beta(\mathbf{L}_{21}\mathbf{U})^-+\beta(\mathbf{L}_{22}\mathbf{D}_L)^-;\mathbf{0}\big] \\ \tilde{\mathbf{D}} & =\mathbf{D}\mathbf{V}^T\mathbf{V}+\lambda\mathbf{D}+\gamma\big(\mathbf{B}^T\mathbf{E}^T\mathbf{E}\bar{\mathbf{B}}\mathbf{D}\mathbf{q}\mathbf{q}^T)^+ +\gamma\big(\bar{\mathbf{B}}^T\mathbf{E}^T\mathbf{E}\mathbf{o}\mathbf{q}^T\big)^- \\ & +\big[\beta(\mathbf{L}_{21}\mathbf{U})^+ +\beta(\mathbf{L}_{22}\mathbf{D}_L)^+ +\eta\big(\mathbf{D}_L\mathbf{p}\mathbf{p}^T\big)^+ +\eta\big(\mathbf{y}_L\mathbf{p}^T\big)^-;\mathbf{0}\big]\end{aligned} \tag{A.5}$$

where for any matrix $\mathbf{X}$, $(\mathbf{X})^+$ and $(\mathbf{X})^-$ denote the positive and negative parts of $\mathbf{X}$, respectively. Specifically, we have $(\mathbf{X})^+=\frac{ABS(\mathbf{X})+\mathbf{X}}{2}$ and $(\mathbf{X})^-=\frac{ABS(\mathbf{X})-\mathbf{X}}{2}$, $ABS(\mathbf{X})$ is the matrix with the absolute value of elements in $\mathbf{X}$.

Update $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{T}$. The partial derivative of the Lagrange objective function w.r.t. $\mathbf{U}$ and updating rule are as follows,

$$\begin{aligned}\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \mathbf{U}} &= \alpha(\mathbf{Y} \odot (\mathbf{U}\mathbf{T}\mathbf{U}^T - \mathbf{A}))\mathbf{U}\mathbf{T}^T + \alpha(\mathbf{Y} \odot (\mathbf{U}\mathbf{T}\mathbf{U}^T - \mathbf{A}))^T\mathbf{U}\mathbf{T} \\ &\quad + \lambda\mathbf{U} - \Psi_U + \beta(\mathbf{L}_{11}\mathbf{U} + \mathbf{L}_{12}\mathbf{D}_L)\end{aligned} \tag{A.6}$$

$$\mathbf{U}_{ij} \leftarrow \mathbf{U}_{ij}\sqrt{\frac{[\hat{\mathbf{U}}](i,j)}{[\tilde{\mathbf{U}}](i,j)}} \tag{A.7}$$

$$\begin{aligned}\hat{\mathbf{U}} &= \alpha(\mathbf{Y} \odot \mathbf{A})\mathbf{U}\mathbf{T}^T + \alpha(\mathbf{Y} \odot \mathbf{A})^T\mathbf{U}\mathbf{T} + \beta(\mathbf{L}_{11}\mathbf{U})^- + \beta(\mathbf{L}_{12}\mathbf{D}_L)^- \\ \tilde{\mathbf{U}} &= \alpha(\mathbf{Y} \odot \mathbf{U}\mathbf{T}\mathbf{U}^T)\mathbf{U}\mathbf{T}^T + \alpha(\mathbf{Y} \odot \mathbf{U}\mathbf{T}\mathbf{U}^T)^T\mathbf{U}\mathbf{T} + \lambda\mathbf{U} \\ &\quad + \beta(\mathbf{L}_{11}\mathbf{U})^+ + \beta(\mathbf{L}_{12}\mathbf{D}_L)^+\end{aligned} \tag{A.8}$$

The partial derivatives of the Lagrange objective w.r.t $\mathbf{V}$ and updating rule are,

$$\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = (\mathbf{D}\mathbf{V}^T - \mathbf{X})^T\mathbf{D} + \lambda\mathbf{V} - \Psi_V \tag{A.9}$$

$$\mathbf{V}_{ij} \leftarrow \mathbf{V}_{ij}\sqrt{\frac{[\mathbf{X}^T\mathbf{D}](i,j)}{[\mathbf{V}\mathbf{D}^T\mathbf{D} + \lambda\mathbf{V}](i,j)}} \tag{A.10}$$

The partial derivative of the Lagrange objective w.r.t $\mathbf{T}$ and the updating rule are,

$$\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \mathbf{T}} = \alpha\mathbf{U}^T(\mathbf{Y} \odot (\mathbf{U}\mathbf{T}\mathbf{U}^T - \mathbf{A}))\mathbf{U} + \lambda\mathbf{T} - \Psi_T \tag{A.11}$$

$$\mathbf{T}_{ij} \leftarrow \mathbf{T}_{ij}\sqrt{\frac{[\alpha\mathbf{U}^T(\mathbf{Y} \odot \mathbf{A})\mathbf{U}](i,j)}{[\alpha\mathbf{U}^T(\mathbf{Y} \odot \mathbf{U}\mathbf{T}\mathbf{U}^T)\mathbf{U} + \lambda\mathbf{T}](i,j)}} \tag{A.12}$$

Update $\mathbf{p}$ and $\mathbf{q}$. Optimization w.r.t $\mathbf{p}$ and $\mathbf{q}$ are essentially least square problems. By setting $\frac{\partial \mathcal{L}}{\partial \mathbf{p}} = 0$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{q}} = 0$, the closed from solutions of $\mathbf{p}$ and $\mathbf{q}$ are as follows,

$$\begin{aligned}\mathbf{p} &= (\eta\mathbf{D}_L^T\mathbf{D}_L + \lambda\mathbf{I})^{-1}\eta\mathbf{D}_L^T\mathbf{y}_L \\ \mathbf{q} &= (\gamma\mathbf{D}^T\bar{\mathbf{B}}^T\mathbf{E}\bar{\mathbf{B}}\mathbf{D} + \lambda\mathbf{I})^{-1}\gamma\mathbf{D}^T\bar{\mathbf{B}}^T\mathbf{E}\mathbf{o}\end{aligned} \tag{A.13}$$

Where $\mathbf{I}$ is an identity matrix, and $\mathbf{E} \in \mathbb{R}^{l\times l}$ with $\mathbf{e}_k, k = 1, \ldots, l$ on the diagonal and zeros everywhere else.

We present the details to optimize TriFN in Algorithm 10. We first randomly initialize $\mathbf{U}, \mathbf{V}, \mathbf{T}, \mathbf{D}, \mathbf{p}, \mathbf{q}$ in line 1, and construct the Laplacian matrix $\mathbf{L}$ in line 2. Then we repeatedly update related parameters through Line 4 to Line 8 until convergence. Finally, we predict the labels of unlabeled news $\mathbf{y}_U$ in line 10. The convergence of Algorithm 10 is guaranteed because the objective function is nonnegative and in each

Input : $\mathbf{X}, \mathbf{A}, \mathbf{B}, \mathbf{W}, \mathbf{Y}, \mathbf{o}, \mathbf{y}_L, \alpha, \beta, \gamma, \lambda, \eta$
Output : $\mathbf{y}_U$

Randomly initialize $\mathbf{U}, \mathbf{V}, \mathbf{T}, \mathbf{D}, \mathbf{p}, \mathbf{q}$;
Precompute Laplacian matrix $\mathbf{L}$;
repeat
    Update $\mathbf{D}$ with Eqn A.4;
    Update $\mathbf{U}$ with Eqn A.8;
    Update $\mathbf{V}$ with Eqn A.10;
    Update $\mathbf{T}$ with Eqn A.12;
    Update $\mathbf{p}$,$\mathbf{q}$ with Eqn A.13;
until *convergence*;
Calculate $\mathbf{y}_U = \text{Sign}(\mathbf{D}_U\mathbf{p})$;

Algorithm 2: The optimization process of TriFN framework

iteration it will monotonically decrease the objective value, and finally it will converge to an optimal point (Lee and Seung, 2001).

The main computation cost comes from the fine-tuning variables for Algorithm 10. In each iteration, the time complexity for computing $\mathbf{D}$ is $\mathcal{O}(nd+nld^2+rd+rm+n^2)$. Similarly, the computation cost for $\mathbf{V}$ is approximately $\mathcal{O}(tnd)$, for $\mathbf{U}$ is $\mathcal{O}(m^4d^3+md)$, for $\mathbf{T}$ is about $\mathcal{O}(m^4d^3 + m^2d^2)$. To update $\mathbf{p}$ and $\mathbf{q}$, the costs are approximately $\mathcal{O}(d^3 + d^2 + dr)$ and $\mathcal{O}(d^2ln + d^3 + dl)$. The overall time complexity is the sum of the costs of initialization and fine-tuning.

APPENDIX B

THE REPRODUCIBILITY FOR DEFEND

In this section, we provide more details of the experimental setting and configuration to enable the reproducibility of dEFEND.

B.0.1 dEFEND for Fake News Detection

We compared the proposed framework dEFEND, with 7 baseline methods, including RST, LIWC, text-CNN, HAN, TCNN-URG, HPA-BLSTM, and CSI. All codes that we have implemented are available under the folder "Fake new detection" through the following link: https://tinyurl.com/ybl6gqrm. Other codes were obtained as follows:

- RST: we used the publicly available implementation for paper (Ji and Eisenstein, 2014): https://github.com/jiyfeng/DPLP
- LIWC: we used the publicly available tool at: http://liwc.wpengine.com/
- text-CNN: we used the publicly available implementation at: https://github.com/dennybritz/cnn-text-classification-tf
- HAN: we used the publicly available implentation at: https://github.com/richliao/textClassifier
- TCNN-URG: we implemented this algorithm based on the description in the paper (Qian *et al.*, 2018), and shared the code, named as *tcnn.py*, in the above link
- HPA-BLSTM: we used the implementation provided by the authors of (Guo *et al.*, 2018a)
- CSI: we used the implementation available at: https://github.com/sungyongs/CSI-Code
- dEFEND: we implemented our algorithm in Python–*defend.py* for main algorithm and *go_ defend.py* for data processing–and shared them in the above link.

For the dataset, we also used a publicly available dataset, FakeNewsNet (Shu *et al.*, 2018b), available at: https://github.com/KaiDMML/FakeNewsNet. For parameter settings for dEFEND, we introduce the details of major parameter setting as shown in Table 16. The descriptions of the major parameters are as follows:

- MAX_SENTENCE_LENGTH: the threshold to control the maximum length of news sentences
- MAX_SENTENCE_COUNT: the threshold to control the maximum count of sentences
- MAX_COMMENT_LENGTH: the threshold to control the maximum length of user comments
- MAX_COMMENT_COUNT: the threshold to control the maximum count of user comments
- Vocabulary Size: the threshold to control the maximum size of vocabulary

Table 16: The details of the parameters of dEFEND

Parameter	PolitiFact	GossipCop
MAX_SENTENCE_LENGTH	120	120
MAX_SENTENCE_COUNT	50	50
MAX_COMMENT_COUNT	150	150
MAX_COMMENT_LENGTH	120	120
Word Embedding	Glove[15]	Glove[16]
Embedding Dimension	100	100
d	100	100
k	80	80
Batch Size	30	20
Maximum Epochs	20	20
Vocabulary Size	20,000	20,000
Learning Rate	0.01	0.001
RMSprop parameter (ρ)	0.9	0.9
RMSprop parameter (ϵ)	1e-8	1e-8
RMSprop parameter (decay)	0	0

- Embedding Dimension: the dimension of embedding layer
- Word Embedding: the word emending package used for initialize the word vectors
- d: the size of hidden states for BLSTM
- k: the size of attention maps as in Eqn. 4.7

B.1 Explainability on News and Comments

We elaborate further details on how we evaluated the explainability of sentences and comments in experiments.

B.1.1 News Sentences

We obtained the ground truth of check-worthy sentences from the online tool, ClaimBuster, with its default setting. ClaimBuster is available at: https://idir-server2.uta.edu/claimbuster/.

B.1.2 User Comments

We introduce the details of the two tasks we deployed at Amazon Mechanical Turk.

- Task 1. We presented each fake article with two lists of top-k comments identified by dEFEND and HPA-LSTM, and let workers choose the list that can "collectively" explain better why this is fake news. In order to remove the position bias, we shuffled the order of the two lists randomly, between top and bottom. Each Human Intelligence Task (HIT) contains five fake articles, and was assigned to three distinct workers. The HIT screen-shot for Task 1 is shown in Figure 20.
  To ensure the quality of crowdsourcing task, we set two requirements for AMT workers: (1) the approved percentage of assignments of a worker should be greater than 95%; and (2) the location of a worker should be in US.
- Task 2. We tested the explainability of user comments detected by dEFEND and HPA-LSTM, respectively. We presented each fake news article with a list of "mixed" comments identified by two methods, and let workers to assign a score of 0-4 to each comment, where 0 means "not explainable at all," 1 means "not much explainable," 3 means "explainable a bit," 4 means "highly explainable," and 2 means "somewhere in between." Note that in order to remove the position bias, we shuffled the order of the comments randomly. Further, to ensure the quality of the task, we applied the same two requirements as in Task 1. Each HIT had one fake article, and was assigned to 3 distinct workers. The HIT screen-shot of Task 2 is shown in Figure 21.

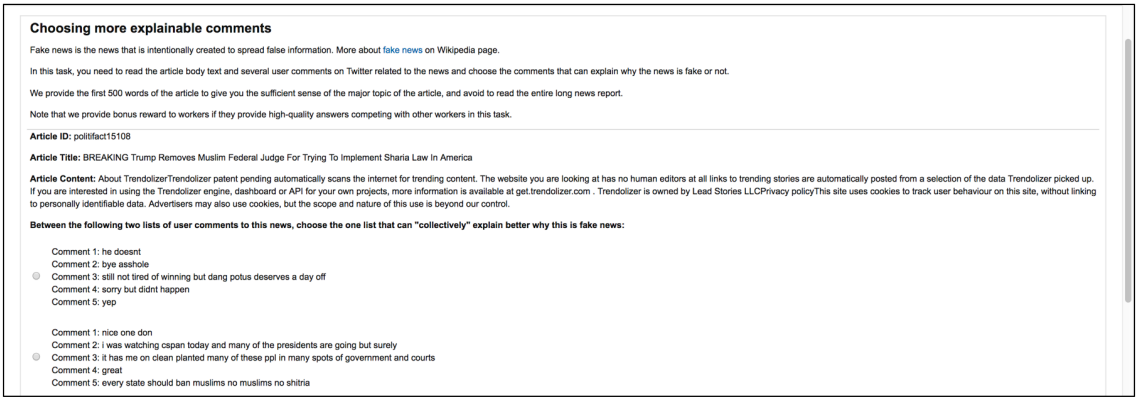


Figure 20: Task 1: Choosing collectively more explainable user comments for fake news articles.

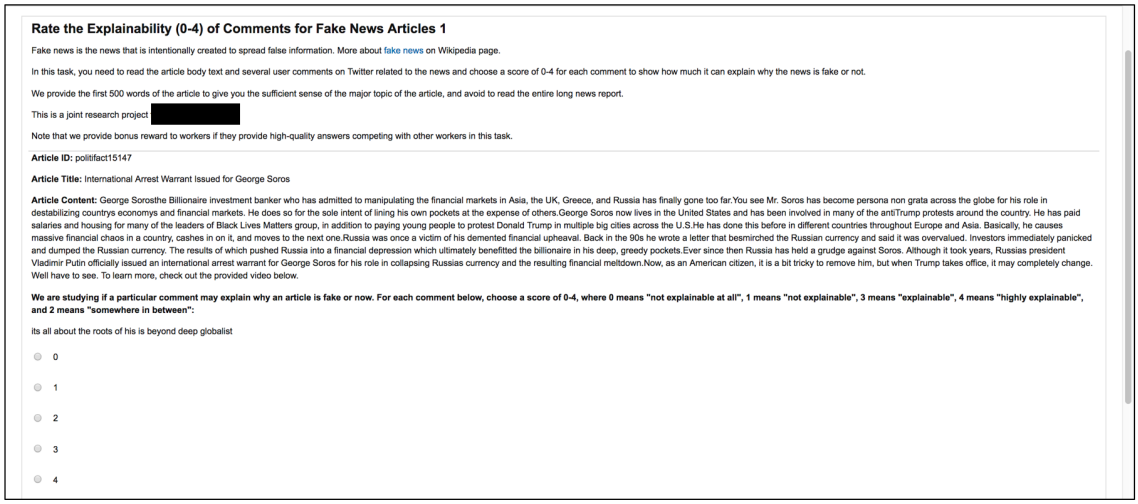


Figure 21: Task 2: Rating the explainability (0-4) of user comments for fake news articles.

APPENDIX C

DATA REPOSITORIES, TOOLS AND ACTIVITIES ON DISINFORMATION RESEARCH

C.1 Data Repository

The first and most important step to detect fake news is to collect a benchmark dataset. Despite several existing computational solutions on the detection of fake news, the lack of comprehensive and community-driven fake news datasets has become one of major roadblocks. In this appendix, we introduce a multi-dimensional data repository *FakeNewsNet*[17], which contains two datasets with news content, social context, and spatiotemporal information (Shu *et al.*, 2018b). For related datasets on fake news, rumors, etc. The readers can refer to several other survey papers such as (Sharma *et al.*, 2019; Zhang and Ghorbani, 2019).

The constructed FakeNewsNet repository has the potential to boost the study of various open research problems related to fake news study. First, the rich set of features in the datasets provides an opportunity to experiment with different approaches for fake new detection, understand the diffusion of fake news in social network and intervene in it. Second, the temporal information enables the study of early fake news detection by generating synthetic user engagements from historical temporal user engagement patterns in the dataset (Qian *et al.*, 2018). Third, we can investigate the fake news diffusion process by identifying provenances, persuaders, and developing better fake news intervention strategies (Shu *et al.*, 2018a). Our data repository can serve as a starting point for many exploratory studies for fake news, and provide a better, shared insight into disinformation tactics. This data repository is continuously updated with new sources and features. For a better comparison of the differences, we list existing popular fake news detection datasets below and compare them with the FakeNewsNet repository in Table 17.

- *BuzzFeedNews*[18]: This dataset comprises a complete sample of news published in Facebook from 9 news agencies over a week close to the 2016 U.S. election from September 19 to 23 and September 26 and 27. Every post and the linked article were fact-checked claim-by-claim by 5 BuzzFeed journalists. It contains 1,627 articles –826 mainstream, 356 left-wing, and 545 right-wing articles.
- *LIAR*[19]: This dataset (Wang, 2017) is collected from fact-checking website PolitiFact. It has 12.8 K human labeled short statements collected from PolitiFact and the statements are labeled into six categories ranging from completely false to completely true as pants on fire, false, barely-true, half-true, mostly true, and true.

[17]https://github.com/KaiDMML/FakeNewsNet

[18]https://github.com/BuzzFeedNews/2016-10-facebook-fact-check/tree/master/data

[19]https://www.cs.ucsb.edu/ william/software.html

- *BS Detector*[20]: This dataset is collected from a browser extension called BS detector developed for checking news veracity. It searches all links on a given web page for references to unreliable sources by checking against a manually compiled list of domains. The labels are the outputs of the BS detector, rather than human annotators.
- *CREDBANK*[21]: This is a large-scale crowd-sourced dataset (Mitra and Gilbert, 2015) of around 60 million tweets that cover 96 days starting from Oct. 2015. The tweets are related to over 1,000 news events. Each event is assessed for credibilities by 30 annotators from Amazon Mechanical Turk.
- *BuzzFace*[22]: This dataset (Santia and Williams, 2018) is collected by extending the BuzzFeed dataset with comments related to news articles on Facebook. The dataset contains 2263 news articles and 1.6 million comments discussing news content.
- *FacebookHoax*[23]: This dataset (Tacchini *et al.*, 2017) comprises information related to posts from the facebook pages related to scientific news (non- hoax) and conspiracy pages (hoax) collected using Facebook Graph API. The dataset contains 15,500 posts from 32 pages (14 conspiracy and 18 scientific) with more than 2,300,000 likes.
- *NELA-GT-2018*[24]: This dataset collects articles between 02/2018-11/2018 from 194 news and media outlets including mainstream, hyper-partisan, and conspiracy sources, resulting in 713k articles. The ground truth labels are integrated from 8 independent assessments.

From Table 17, we observe that no existing public dataset can provide all possible features of news content, social context, and spatiotemporal information. Existing datasets have some limitations that we try to address in our data repository. For example, BuzzFeedNews only contains headlines and text for each news piece and covers news articles from very few news agencies. LIAR dataset contains mostly short statements instead of entire news articles with the meta attributes. BS Detector data is collected and annotated by using a developed news veracity checking tool, rather than using human expert annotators. CREDBANK dataset was originally collected for evaluating tweet credibilities and the tweets in the dataset are not related to

[20]https://github.com/bs-detector/bs-detector

[21]http://compsocial.github.io/CREDBANK-data/

[22]https://github.com/gsantia/BuzzFace

[23]https://github.com/gabll/some-like-it-hoax

[24]https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ULHLCB

Table 17: The comparison with representative fake news detection datasets

Datasets	News Content		Social Context				Spatiotemporal	
	Linguistic	Visual	User	Post	Response	Network	Spatial	Temporal
BuzzFeedNews	✓							
LIAR	✓							
BS Detector	✓							
CREDBANK	✓		✓	✓			✓	✓
BuzzFace	✓			✓	✓			✓
FacebookHoax	✓		✓	✓	✓			
NELA-GT-2018	✓							
FakeNewsNet	✓	✓	✓	✓	✓	✓	✓	✓

the fake news articles and hence cannot be effectively used for fake news detection. BuzzFace dataset has basic news contents and social context information but it does not capture the temporal information. The FacebookHoax dataset consists very few instances about the conspiracy theories and scientific news.

To address the disadvantages of existing fake news detection datasets, the proposed FakeNewsNet repository collects multi-dimension information from news content, social context, and spatiotemporal information from different types of news domains such as political and entertainment sources.

C.1.0.0.1 Data Integration

In this part, we introduce the dataset integration process for the FakeNewsNet repository. We demonstrate in Figure 22 how we can collect news contents with reliable ground truth labels, and how we obtain additional social context and spatialtemporal information.

News Content To collect reliable ground truth labels for fake news, we utilize fact-checking websites to obtain news contents for fake news and true news such as *PolitiFact*[25] and *GossipCop*[26].

In PolitiFact, journalists and domain experts review the political news and provide

[25]https://www.politifact.com/

[26]https://www.gossipcop.com/

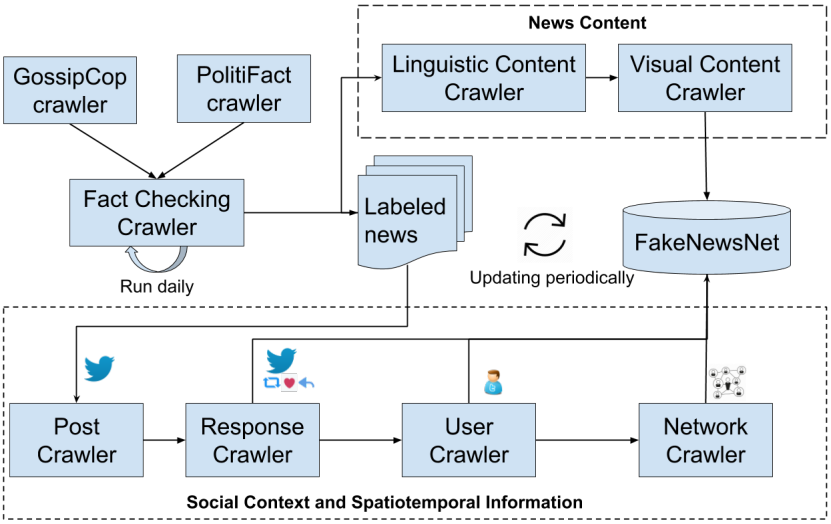


Figure 22: The flowchart of data integration process for FakeNewsNet. It mainly describes the collection of news content, social context and spatiotemporal information.

fact-checking evaluation results to claim news articles as fake[27] or real[28]. We utilize these claims as ground truths for fake and real news pieces. In PolitiFact's fact-checking evaluation result, the source URLs of the web page that published the news articles are provided, which can be used to fetch the news contents related to the news articles. In some cases, the web pages of source news articles are removed and are no longer available. To tackle this problem, we i) check if the removed page was archived and automatically retrieve content at the Wayback Machine[29]; and ii) make use of Google web search in automated fashion to identify news article that is most related to the actual news.

GossipCop is a website for fact-checking entertainment stories aggregated from various media outlets. GossipCop provides rating scores on the scale of 0 to 10 to classify a news story as the degree from fake to real. In order to collect true entertainment news pieces, we crawl the news articles from E! Online[30], which is a well-known trusted media website for publishing entertainment news pieces. We consider all the articles from E! Online as real news sources. We collect all the news stories from GossipCop with rating scores less than 5 as the fake news stories. Since

[27]available at https://www.politifact.com/subjects/fake-news/

[28]available at https://www.politifact.com/truth-o-meter/rulings/true/

[29]https://archive.org/web/

[30]https://www.eonline.com/

GossipCop does not explicitly provide the URL of the source news article, so we search the news headline in Google or the Wayback Machine archive to obtain the news source information.

Social Context: The user engagements related to the fake and real news pieces from fact-checking websites are collected using search API provided by social media platforms such as the Twitter's Advanced Search API[31]. The search queries for collecting user engagements are formed from the headlines of news articles, with special characters removed from the search query to filter out the noise. After we obtain the social media posts that directly spread news pieces, we further fetch the user *response* towards these posts such as replies, likes, and reposts. In addition, when we obtain all the users engaging in news dissemination process, we collect all the metadata for user profiles, user posts, and the social network information.

Spatiotemporal Information: The spatiotemporal information includes spatial and temporal information. For spatial information, we obtain the locations explicitly provided in user profiles. The temporal information indicates that we record the timestamps of user engagements, which can be used to study how fake news pieces propagate on social media, and how the topics of fake news are changing over time. Since fact-checking websites periodically update newly coming news articles, so we dynamically collect these newly added news pieces and update the FakeNewsNet repository as well. In addition, we keep collecting the user engagements for all the news pieces periodically in the FakeNewsNet repository such as the recent social media posts, and second order user behaviors such as replies, likes, and retweets. For example, we run the news content crawler and update Tweet collector per day. The spatiotemporal information provides useful and comprehensive information for studying fake news problem from a temporal perspective.

C.2 Tools

In this appendix, we introduce some representative online tools for tracking and detecting fake news on social media.

[31]https://twitter.com/search-advanced?lang=en

C.2.0.0.1 Hoaxy

Hoaxy[32] aims to build a uniform and extensible platform to collect and track misinformation and fact-checking (Shao *et al.*, 2016), with visualization techniques to understand the misinformation propagation on social media.

Data Scraping The major components included a tracker for the Twitter API, and a set of crawlers for both fake news and fact checking websites and databases (see Figure 23).

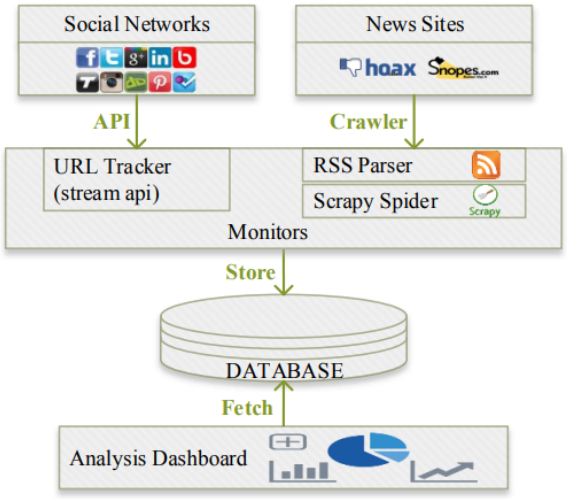


Figure 23: The framework of Hoaxy.

The system collects data from two main sources: news websites and social media. From the first group we can obtain data about the origin and evolution of both fake news stories and their fact checking. From the second group we collect instances of these news stories (i.e., URLs) that are being shared online. To collect data from such disparate sources, different technologies are used: Web scraping, Web syndication, and, where available, APIs of social networking platforms. To collect data on news stories they use RSS, which allows a unified protocol instead of manually adapting our scraper to the multitude of Web authoring systems used on the Web. RSS feeds contain information about updates made to news stories. The data is collected from news sites using the following two steps: when a new website is added to our list of monitored sources, a 'deep' crawl of its link structure is performed using a custom

[32]https://hoaxy.iuni.iu.edu/

Python spider written with the Scrapy framework; at this stage, the URL of the RSS feed is identified if it is available. Once all existing stories have been acquired, a 'light' crawl is performed every two hour by checking its RSS feed only. To perform the 'deep' crawl, we use a depth first strategy. The 'light' crawl is instead performed using a breadth-first approach.

Analysis Dashboard Hoaxy provides various visualization interfaces to demonstrate the news spreading process. As shown in Figure 24, we demonstrate the major functionalities on the analysis dashboard. On the top, users can search any news articles by providing specific keywords. On the left side, it demonstrates the the temporal trendiness of the user engagements for the news articles. On the right side, it illustrates the propagation network on Twitter, which clearly convey the information on who spreads the news tweets from whom. In addition, they also evaluate the bot score for all users with BotMeter (Davis *et al.*, 2016).

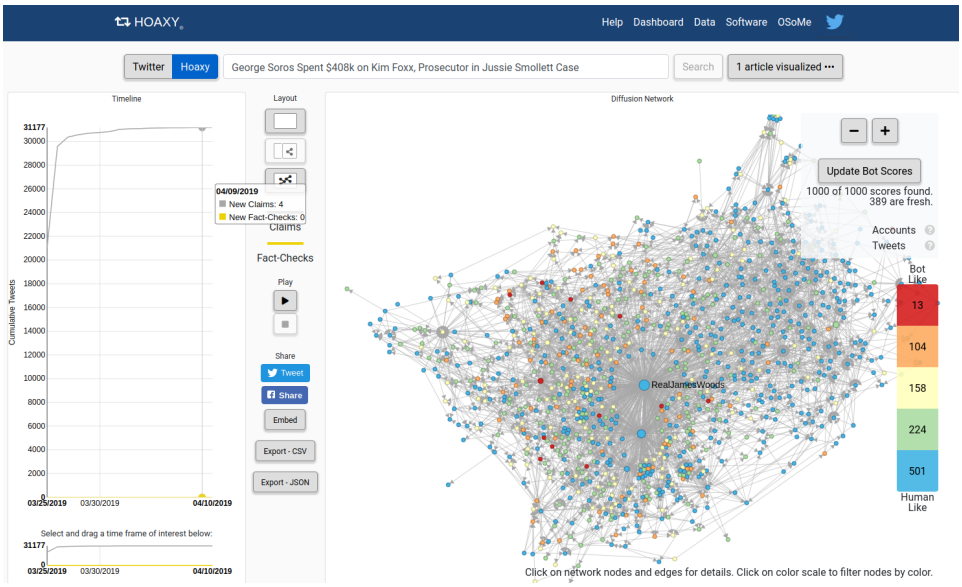


Figure 24: The main dashboard of Hoaxy website.

C.2.0.0.2 FakeNewsTracker

FakeNewsTracker[33] is a system for fake news data collection, detection, and visualization on social media (Shu *et al.*, 2019b). It mainly consists of the following

[33]http://blogtrackers.fulton.asu.edu:3000/

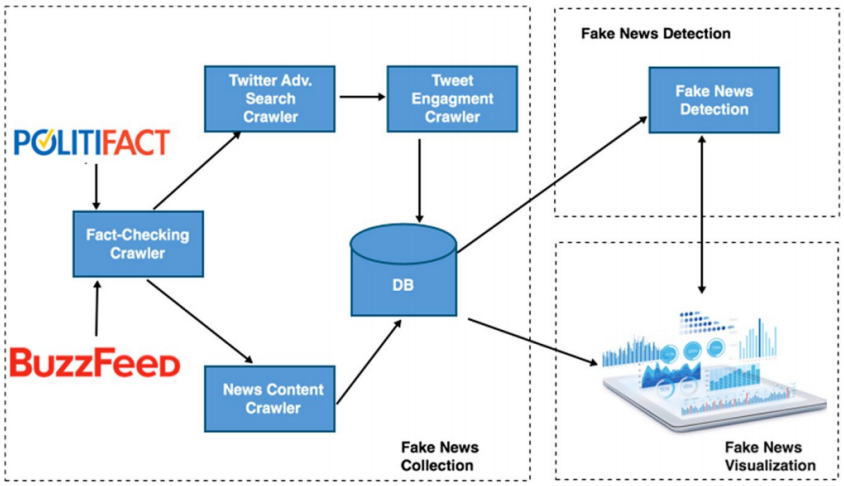


Figure 25: The framework of FakeNewsTracker.

components (see Figure 25): 1) fake news collection; 2) fake news detection; and 3) fake news visualization.

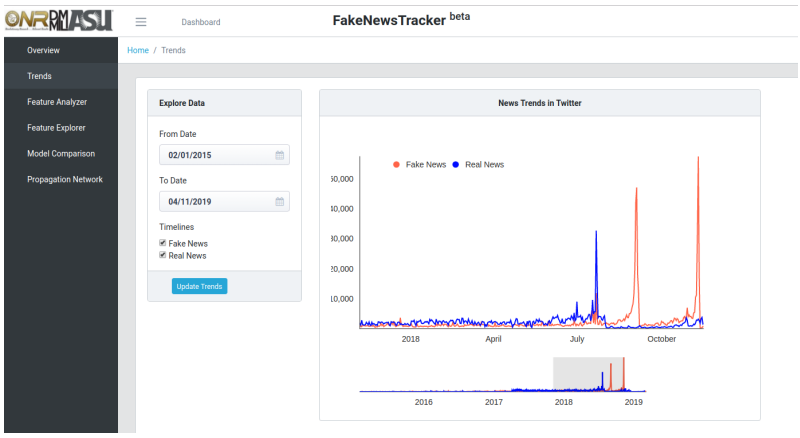
Collecting Fake News Data Fake news is widely spread across various online platforms. We use some of the fact-checking websites like PolitiFact as a source for collecting fake news information. On these fact-checking sites, fake news information is provided by the trusted authors and relevant claims are made by the authors on why the mentioned news is not true. The detailed collection procedure is described in Figure 22.

Detecting Fake News A deep learning model is proposed to learn neural textual features from news content, and temporal representations from social context simultaneously to predict fake news. An auto-encoder (Kingma and Welling, 2013) is used to learn the feature representation of news articles, bu reconstructing the news content, and LSTM is utilized to learn the temporal features of user engagements. Finally, the learned feature representations of news and social engagements are fused to predict fake news.

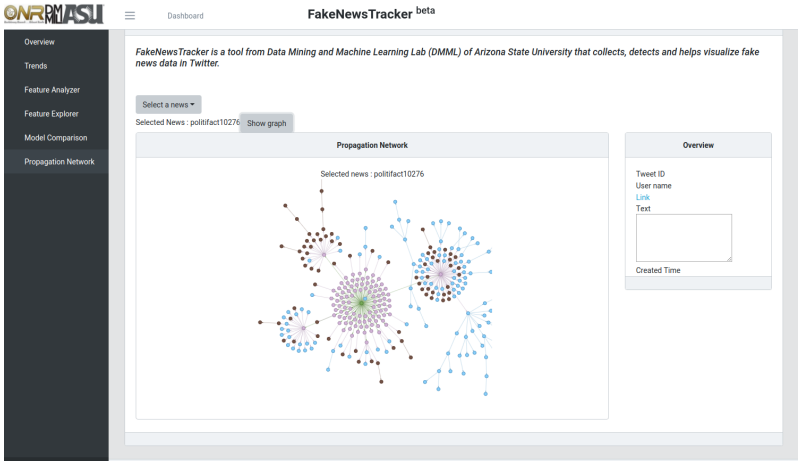
Visualization Fake News in Twitter We have developed a fake news visualization as shown in Figure 26 for the developing insights on the collected data through various interfaces. We demonstrate the temporal trends of the number of tweets spreading fake and real news in a specific time period, as in Figure 26a.

In addition, we can explore the social network structure among users in the propagation network (see Figure 26b for an example), and further compare the differences between the users who interact with the fake news and the true news.

For identifying the differences in the news content of the true news and the fake news we have used word cloud representation of the words for the textual data. We



(a) User Interface of Trend on News Spreading.



(b) User Interface on News Propagation Networks.

Figure 26: Demonstration of FakeNewsTracker system.

search for fake news within a time frame and identify the relevant data. In addition, we provide the comparison of feature significance and model performance as part of this dashboard. Moreover, we could see how fake news is spread around certain areas using the geo-locations of tweets.

C.2.0.0.3 dEFEND

dEFEND[34] is a fake news detection system that are also able to provide explainable user comments on Twitter. dEFEND (see Figure 27) mainly consists of two major components: a web-based user interface and a backend which integrates our fake news detection model.

The web-based interface provides users with explainable fact-checking of news. A user can input either the tweet URL or the title of the news. A screenshot was shown in Figure 28. On typical fact-checking websites, a user just sees the check-worthy score of news (like Gossip Cop[35]) or each sentence (like ClaimBuster[36]). In our approach, the user can not only see the detection result (in the right of Figure 28a), but also can find all the arguments that support the detection result, including crucial sentences in the article (in the middle of Figure 28b) and explainable comments from social media platforms (in the right of Figure 28b). At last, the user can also review the results and find related news/claims.

The system also provides exploratory search functions including news propagation network, trending news, top claims and related news. The news propagation network (in the left of Figure 28b) is to help readers understand the dynamics of real and fake news sharing, as fake news are normally dominated by very active users, while real news/fact checking is a more grass-roots activity (Shao *et al.*, 2016). Trending news, top claims and related news (in the lower left of Figure 28a) can give some query suggestions to users.

The backend consists of multiple components: (1) a database to store the pre-trained results as well as a crawler to extract unseen news and its comments, (2) the dEFEND algorithm module based on explainable deep learning fake news detection, which gives the detection result and explanations simultaneously and (3) an exploratory component that shows the propagation network of the news, trending and related news.

Exploratory Search The system also provides users with browsing functions. Consider a user who doesn't know what to check specifically. By browsing the trending news, top claims and news related to the previous search right below the input box, the user can get some ideas about what he could do. News can be the coverage of an event, such as "Seattle Police Begin Gun Confiscations: No Laws Broken, No Warrant,

[34]http://fooweb-env.qnmbmwmxj3.us-east-2.elasticbeanstalk.com/

[35]https://www.gossipcop.com/

[36]https://idir-server2.uta.edu/claimbuster/

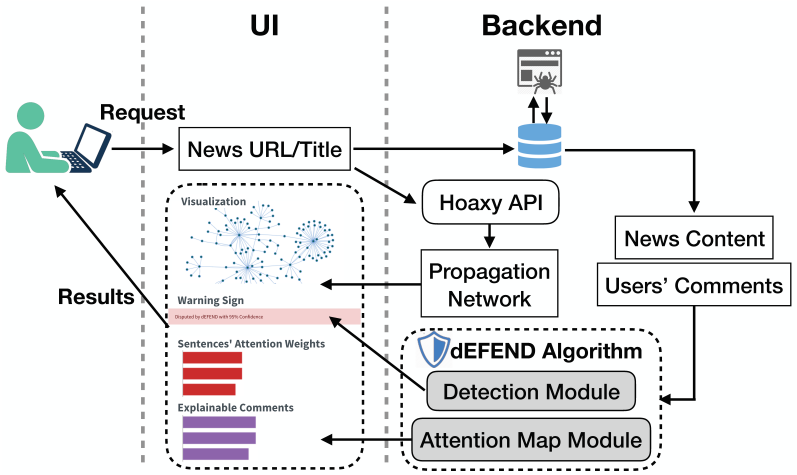
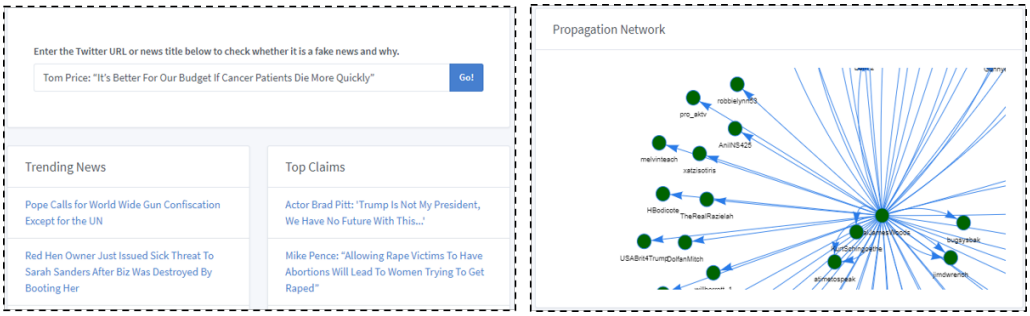


Figure 27: The framework of dEFEND.



(a) User Interface of Search: the input box (upper left), query suggestions (lower left) and intuitive propagation network (right).

(b) Explainable Fact Checking: news content (left), explainable sentences (upper right) and comments (lower right).

Figure 28: Demonstration of dEFEND system.

No Charges" and claim is the coverage around what a celebrity said, such as "Actor Brad Pitt: 'Trump Is Not My President, We Have No Future With This..."'. Users can search these titles by clicking on them. The news related to the user's previous search

is recommended. For example, news "Obama's Health Care Speech to Congress" is related to the query "It's Better For Our Budget If Cancer Patients Die More Quickly".

Explainable Fact-checking Consider a user who wants to check whether Tom Price has said "It's Better For Our Budget If Cancer Patients Die More Quickly". The user first enters the tweet URL or the title of a news in the input box in Figure 28a. The system would return the check-worthy score, the propagation network, sentences with explainable scores, and comments with explainable scores to the user in Figure 28b. The user can zoom in the network to check the details of the diffusion path. Each sentence is shown in the table along with its score. The higher the score, the more likely the sentence contains check-worthy factual claims. The lower the score, the more non-factual and subjective the sentence is. The user can sort the sentences either by the order of appearance or by the score. Comments' explainable scores are similar to sentences'. The top-*5* comments are shown in the descending order of their explainable score.

C.2.0.0.4 NewsVerify

NewsVerify[37] is a real-time news verification system which can detect the credibility of an event by providing some keywords about it (Zhou *et al.*, 2015).

NewsVerify mainly contains three stages: 1) crawling data; 2) building an ensemble model; and 3) visualizing the results. Given the keywords and time range of a news event, the related microblogs can be collected through the search engine of Sina Weibo. Based on these messages, the key users and microblogs can be extracted for further analysis. The key users are used for information source certification while the key microblogs are used for propagation and content certification. All the data above are crawled through distributed data acquisition system which will be illustrated below. After three individual models have been developed, the scores from the above mentioned models are combined via weighted combination. Finally, an event level credibility score is provided, and each single model will also have a credibility score that measure the credibility of corresponding aspect. To improve the user experience of our application, the results are visualized from various perspectives, which provide useful information of events for further investigation.

Data Acquisition Three kinds of information are collected: microblogs, propagation and microbloggers. Like most distributed system, NewsVerify also has master node and child nodes. The master node is responsible for task distribution and results integration while child node process the specific task and store the collected data in

[37]https://www.newsverify.com/

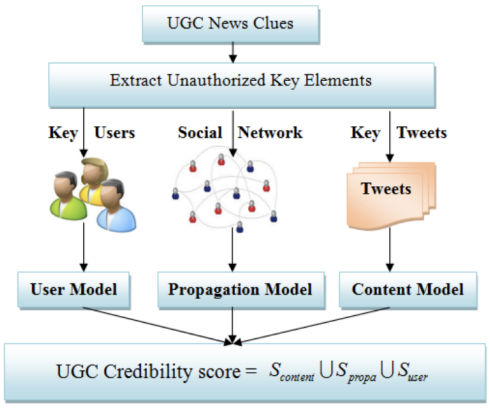


Figure 29: The framework of NewsVerify system.

the appointed temporary storage space. The child node will inform the master node after all tasks finished. Then, master node will merge all slices of data from temporary storage space and stored the combined data in permanent storage space. After above operations, the temporary storage will be deleted. The distributed system is based on ZooKeeper[38], a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. As the attributes of frequent data interaction,stored,read, we adopt efficient key-val database Redis to handle the real-time data acquisition task. Redis, works with an in-memory dataset, can achieve outstanding performance.

Model Ensemble Different individual models are built to verify the truthfulness of news pieces from the perspective of news content, news propagation, and information source (See Figure 29). The *content-based* model is based on hierarchical propagation networks (Jin *et al.*, 2014). The credibility network has three layers: message layer, sub-event layer and event layer. Following that, the semantic and structure features are exploited to adjust the weights of links in the network. Given a news event and its related microblogs, sub-events are generated by clustering algorithm. Sub-event layer is constructed to capture implicit semantic information within an event. Four types of network links are made to reflect the relation between network nodes. The intra-level links(Message to Message, Sub-event to Sub-event) reflect the relations among entities of a same type while the inter level links(Message to Sub-event,Sub-event to Event) reflect the impact from level to level. After the network constructed, all entities are initialize with credibility values using classification results. We formulate this propagation as a graph optimization problem and provides a global optimal solution to it. The *propagation-based* model propose to compute a propagation influence score over time to capture the temporal trends. The *information source based* model utilize

[38]http://zookeeper.apache.org/

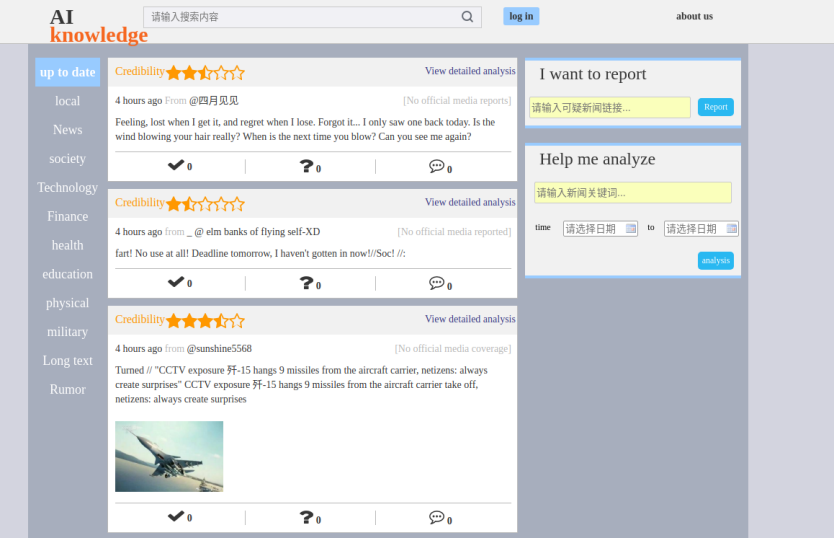


Figure 30: The Interface of NewsVerify System.

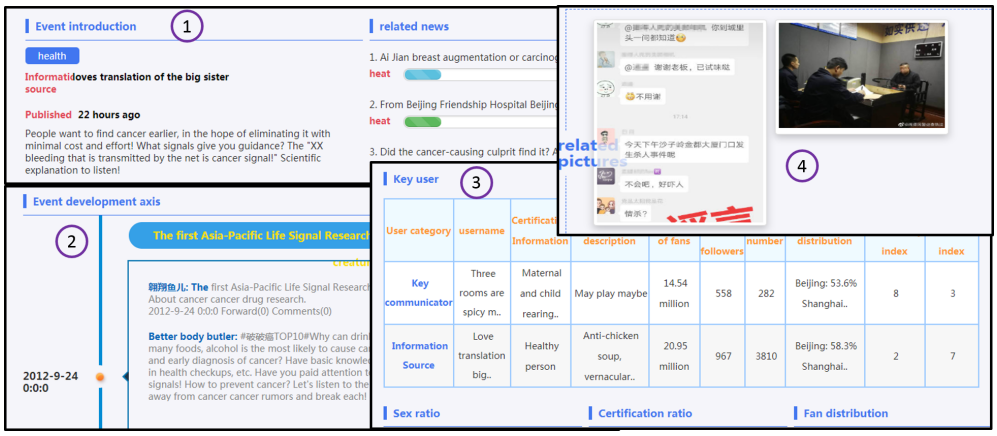


Figure 31: Demonstration of Detail News Analysis of NewsVerify system.

the sentiment and activeness degree as features to help predict fake news. From the aforementioned models, individual score is obtained. Then a weighted logistic regression model can be used to ensemble the result and produce an overall score for the news piece.

Interface Visualization Figure 30 illustrate the interface of NewsVerify system. It allows users to report fake news, and search specific news to verify by providing keywords to the system. It also automatically show the degree of veracity for Weibo

data of different categories. For each Weibo in the time-line, NewsVerify shows the credibility score to justify how likely the Weibo is related to fake news. In addition, it allows interested users to click "View detailed analysis" to learn more about the news. As shown in Figure 31, it mainly demonstrates: (1) the introduction of the news event including the time, source, related news, etc; (2) the changes of trends and topics over time related to the news events; (3) the profiles and aggregated statistics of users who are engaged in the news spreading process such as the key communicator, sex ratio, certification ratio; and (4) the images or videos related to the news events.

C.3 Relevant Activities

Fake news has been attracting interests from experts and practitioners of multiple disciplines. Relevant activities are organized to advance the detection and mitigation of fake news. Specifically, we introduce these efforts in three general categories: *Educational Programs*, *Computational Competitions*, and *Research Workshops and Tutorials*.

C.3.0.0.1 Educational Programs

These programs aim to help design and train interested people about how to identify fake news. The educational programs include handbooks (Murphy, 2019; Ireton and Posetti, 2018), interactive games. For example, in (Ireton and Posetti, 2018), the researchers from UNESCO[39] build a series of curricula and handbooks for journalism education and training. Similarly, a cookbook is built to help identify fake news from the perspectives of transparency, engagement, education, and tools[40]. Interactive online programs are designed that encode some heuristic features of fake news detection to help learn common tricks of identifying fake news via game playing. "Bad News"[41] is an online game that allows users to act as a fake news creator to build a fake "credibility" step by step.

[39]The United Nations Educational, Scientific and Cultural Organization

[40]https://newscollab.org/best-practices/

[41]https://getbadnews.com/

C.3.0.0.2 Computational Competitions

To encourage researchers or students to build computational algorithms to address fake news problems, several competitions are organized online or in conjunction with some conferences. For example, the fake news challenge[42] aims to detect the stance of pairs of headline and body text, which attracts many researchers to create effective solutions to improve the performance (Shang *et al.*, 2018; Riedel *et al.*, 2017). As another example, Bytedance organized a fake news classification challenge in conjunction with the ACM WSDM conference[43], aiming to identify if a given news pieces is related to another piece of fake news. Moreover, the SBP competition on disinformation is regularly held to encourage researchers to combat fake news[44].

C.3.0.0.3 Research Workshops and Tutorials

To bring researchers and practitioners together to brainstorm novel ways of dealing with fake news, different research workshops and tutorials are held from various perspectives. One of the earliest and influential workshop (Mele *et al.*, 2017) aims to define the foundations, actions, and research directions on combating fake news. The social cyber-security working group[45] has brought together experts to deal with various cyber security threats on social media including disinformation and fake news. Recently, the National Academy of Sciences Colloquia hosted a panel on the communication and science of misinformation and fake news[46]. Several tutorials are offered in conjunction with top-tier conferences such as ACM KDD 2019[47], ACM

[42]http://www.fakenewschallenge.org/

[43]https://www.kaggle.com/c/fake-news-pair-classification-challenge/

[44]http://sbp-brims.org/2019/challenge/challenge2_Disinformation.html

[45]https://sites.google.com/view/social-cybersec/

[46]http://www.cvent.com/events/advancing-the-science-and-practice-of-science-communication-misinformation-about-science-in-the-publ/event-summary-c4d9df4d8baf4567ab82042e4f4efb78.aspx

[47]https://www.fake-news-tutorial.com/

WSDM 2019 (Zhou *et al.*, 2019b), AAAI 2018[48], IEEE ICDM 2017[49]. For example, the tutorials of KDD 2019 and WSDM 2019 focus on the fundamental theories, detection strategies, and open issues, the AAAI 2018 tutorial discusses fake news from artificial intelligence and database perspectives, and the ICDM 2017 tutorial presents the detection and spreading patterns of misinformation on social media.

[48]https://john.cs.olemiss.edu/ nhassan/file/aaai2018tutorial.html

[49]http://www.public.asu.edu/ liangwu1/ICDM17MisinformationTutorial.html

BIOGRAPHICAL SKETCH

Kai Shu is a PhD candidate of Computer Science and Engineering at Arizona State University. His research lies in machine learning, data mining, social computing, and applications in disinformation, education, healthcare. He is the leading author of a monograph, Detecting Fake News on Social Media, Morgan & Claypool Publishers, and the leading editor of a book, Disinformation, Misinformation and Fake News in Social Media, Springer Press. He was awarded the ASU CIDSE Doctoral Fellowship 2015 and 2020, and the 1st place of SBP Disinformation Challenge 2018. He co-presented two tutorials in KDD 2019 and WSDM 2019, and has published more than 40 innovative works in highly ranked journals and top conference proceedings such as ACM KDD, SIGIR, WSDM, WWW, CIKM, ECML-PKDD, IEEE ICDM, IJCAI, and AAAI. He interned at Microsoft Research, Yahoo! Research, and HP Labs. More can be found at http://www.public.asu.edu/∼skai2.