DISCO: Comprehensive and Explainable Disinformation Detection

Dongqi Fu
University of Illinois at Urbana-Champaign
Illinois, USA
dongqif2@illinois.edu

Yikun Ban
University of Illinois at Urbana-Champaign
Illinois, USA
yikunb2@illinois.edu

Hanghang Tong
University of Illinois at Urbana-Champaign
Illinois, USA
htong@illinois.edu

Ross Maciejewski
Arizona State University
Arizona, USA
rmacieje@asu.edu

Jingrui He
University of Illinois at Urbana-Champaign
Illinois, USA
jingrui@illinois.edu

ABSTRACT

Disinformation refers to false information deliberately spread to influence the general public, and the negative impact of disinformation on society can be observed in numerous issues, such as political agendas and manipulating financial markets. In this paper, we identify prevalent challenges and advances related to automated disinformation detection from multiple aspects and propose a comprehensive and explainable disinformation detection framework called DISCO. It leverages the heterogeneity of disinformation and addresses the opaqueness of prediction. Then we provide a demonstration of DISCO on a real-world fake news detection task with satisfactory detection accuracy and explanation. The demo video [1] and source code [2] of DISCO is now publicly available. We expect that our demo could pave the way for addressing the limitations of identification, comprehension, and explainability as a whole.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; • **Information systems** → **Document representation**; • **Mathematics of computing** → **Graph algorithms**.

KEYWORDS

Disinformation Detection; Model Explanation; Graph Augmentation

ACM Reference Format:
Dongqi Fu, Yikun Ban, Hanghang Tong, Ross Maciejewski, and Jingrui He. 2022. DISCO: Comprehensive and Explainable Disinformation Detection. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://drive.google.com/file/d/1Nhw1veqjIN9SBz1RLJPDTRVTHuknfjHl
[2] https://github.com/DongqiFu/DISCO

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Disinformation (e.g., fake news) is fabricated to mislead the general public. Historically, disinformation campaigns often took the form of government propaganda with edited newsreels, and the cost of creation and distribution required a funded and coordinated effort. However, with the recent rise of accessible and low-cost computational methods for text manipulation and the broad access to public information channels via the worldwide web, society now finds itself inundated with disinformation that is resulting in large-scale negative societal impacts. Negative effects include but are not limited to, fake news deliberately misleads readers to accept false or biased information for further political agendas or manipulate financial markets; furthermore, fake news also downgrades the credibility of real news and hinders people's ability to distinguish factual information from disinformation [20].

Two main challenges hinder disinformation-related research. Firstly, a characteristic of the spread and creation of disinformation is the co-existence of multiple types of heterogeneous features, which introduces ambiguous factors that could potentially camouflage disinformation from real information. To be more specific, even a single word can have different semantic meanings in different contexts [17, 18]. For example, "Apple" in the food corpus means a kind of fruit, while it stands for the company in the high-tech corpus. Secondly, interpretability is another key to understanding the logic of prediction or classification systems to further build human-machine trust in results and improve detection accuracy [19]. While pioneering state-of-the-art algorithms have been proposed to detect disinformation, many of these models are black-box in nature and lack interpretable mechanisms for explaining why information has been flagged as false. Take fake news as an example, not every sentence in fake news is false. It is critical to specify guidelines for improving the existing model in an explainable way. For instance, to distinguish the importance of different sentences in determining the detection decision [19]. As such, there is a critical need for the new technology that can support disinformation detection both accurately and efficiently (at an early stage before widespread propagation), and in a way that is understandable by both domain experts and the general public.

Contribution. The main contributions of the paper can be summarized as follows. (1) We propose a comprehensive computational

framework for disinformation, named DISCO. It models the heterogeneity of disinformation with graph machine learning techniques, such that the heterogeneity can be positively leveraged to solve challenges for detection and explanation. To be specific, *for the heterogeneity*, DISCO leverages large-scale pre-trained (from general corpus) language models in a transfer learning manner, which means involving a simple projection head could make DISCO achieve high detection accuracy in specific domains like politics news; *for the explanation*, DISCO involves dynamics in a recently proposed graph neural network model [9] such that the word and sentence importance towards the detection variance can be figured out by graph augmentations (i.e., masking nodes and edges in article graphs). (2) We develop an online demo for visualizing the detection and the explanation result of DISCO. (3) We design the real-world experiment with 48,000+ fake and real news articles, and DISCO could outperform than baseline algorithms.

Demonstration. The online demo of DISCO is programmed by Python and allows open interactions with users. The functions of the demo include (1) output the real and fake probabilities of a piece of suspect information; (2) output the misleading degree of each word in the input text and their rankings. A user-guide introduction video of the DISCO demo is also online now.

Related Work. In current disinformation detection tools [6, 15, 16, 25, 26], the query of suspicious information is usually a short phrase (e.g., "Microsoft is a Chinese company") and heavily relies on the retrieval of sufficient background articles to give a fair verdict to that query. To achieve the verification of long articles, the challenges include but not limited to the cost of searching time and searching scope, i.e., a long article may have many candidates from different aspects or searching an emerging disinformation may return a limited number of viable background articles. Our DISCO transfers the searching process into the question comprehension (i.e., via geometric feature extraction and neural detection), such that the open-end queries can be answered in the real time. Also, our model is built upon the large corpus, which has 3 million distinct words from numerous articles, can provide comprehensive knowledge for down-streaming specific domains with limited information. Among the above-mentioned demos [6, 15, 16, 25, 26], our demo is most similar with the online module of [26], which takes an open-end query and give the credibility analysis. Additionally, our DISCO also gives an interpretation in terms of each word's contribution (positive or negative) towards the ground-truth prediction.

2 SYSTEM ARCHITECTURE

The online demo of DISCO has two main parts, front-end, and back-end, as shown in Figure 1. The front-end (1) accepts and passes the user open query (i.e., a suspect piece of information) to the back-end, and (2) receives and shows the detection decision and misleading words rankings from the back-end. The back-end is supported by the graph machine learning techniques, which is responsible for identifying the input information and making the corresponding explanation by (1) building the word graph for the input article (2) extracting the geometric feature of that graph; (3) predicting whether the input article is real or fake; and (4) ranking each word in that text based on the misleading degree. Then the back-end returns all results to the front-end for users. The theoretical details

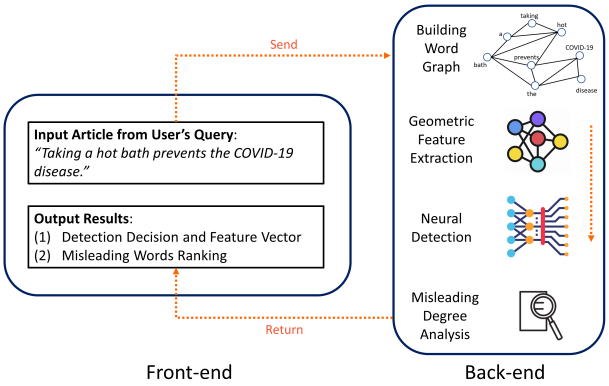

Figure 1: System Architecture of DISCO.

of how these four sequential functions inside the back-end get realized are discussed in the next section.

3 THEORETICAL DETAILS

The back-end side of our DISCO demo is based on graph machine learning techniques, such that an article can be represented as an embedding vector, and the disinformation detection problem is converted into a graph classification problem. First of all, an input article is modeled by a word graph (Subsection 3.1). Second, the entire graph is represented by an embedding vector (Subsection 3.2). Third, the graph-level embedding vector goes into a neural network to get the predictions in terms of the fake news probability and real news probability (Subsection 3.3). Fourth, each word in the article (i.e., each node in the built word graph) is masked under the same graph topological constraint to see each word's contribution towards the input article prediction (Subsection 3.4).

3.1 Building Word Graphs

To build a word graph G (i.e., upper right corner in Figure 1) for an input news article, we follow [13] such that each word in the article stands for a unique node and an edge is established if two words co-occur in a window of k text units. In our demonstration, we set $k = 3$. Take "I eat an apple" as an example, and then the edges could be {I-eat, I-an, eat-an, eat-apple, an-apple} with stop words kept and edges undirected. Beyond [13], we assign each node i in graph G with its own node feature vector $\boldsymbol{x}_i \in \mathbb{R}^{1\times d}$, and $\boldsymbol{x}_i$ that can be retrieved from the word embedding vector of the large-scale pre-trained NLP model, like Bert [7] or Word2Vec [14]. In our demonstration, we adopt a large-scale pre-trained Word2Vec from Google [3] that is pre-trained on a 3 million distinct words corpus, and each word embedding vector is 300-dimensional (i.e., $d = 300$).

3.2 Geometric Feature Extraction

Suppose there are n different words in an article, then we can construct a word graph G with n nodes as mentioned above. Given the input node feature matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ (i.e., $X(i,:) = \boldsymbol{x}_i$), the node hidden representation vector $\boldsymbol{h}_i \in \mathbb{R}^{1\times d}$, $i \in \{1,\ldots,n\}$, can be obtained by

$$\boldsymbol{h}_i = \boldsymbol{p}_i^\top \boldsymbol{X} \tag{1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^{n\times 1}$ is the personalized PageRank vector with node i as the seed node and can be expressed as follows

$$\boldsymbol{p}_i = \alpha \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{p}_i + (1-\alpha)\boldsymbol{r} \tag{2}$$

[3] https://code.google.com/archive/p/word2vec/

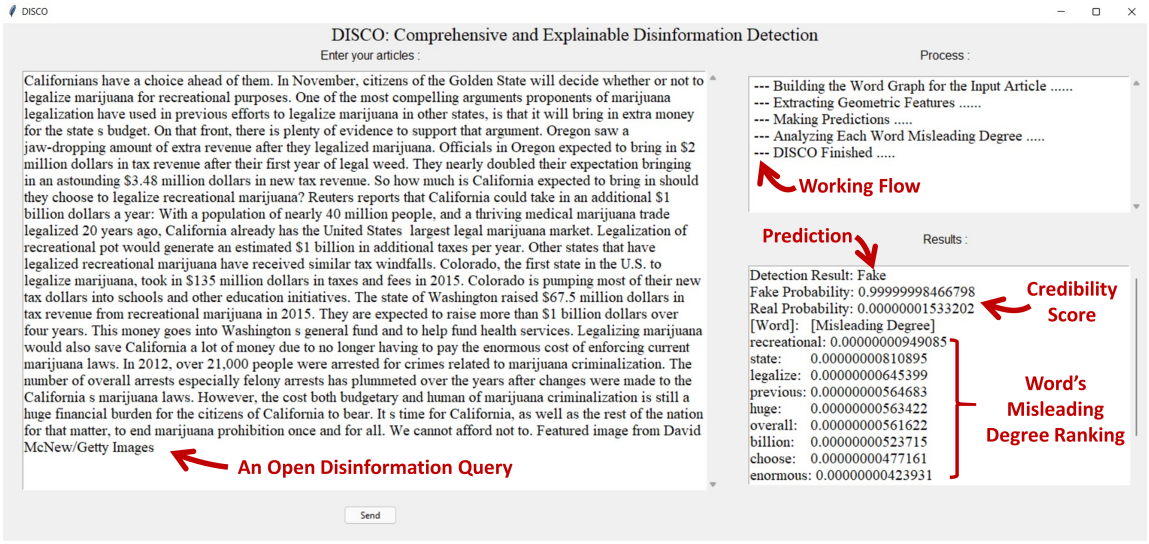

Figure 2: User Interface of DISCO. The left-hand side is the text area to receive open queries from users, the upper right-hand side is the real-time process monitor of DISCO, and the lower right-hand side is the prediction and explanation result.

where $\boldsymbol{A}$ and $\boldsymbol{D}$ are adjacency and degree matrices of word graph G, $\alpha \in [0, 1]$ is the teleportation probability (e.g., $\alpha = 0.85$ in our demo), and $\boldsymbol{r} \in \mathbb{R}^{n\times 1}$ is the personalized vector with $r(i) = 1$ and other entries are 0s.

With this geometric feature extraction, the heterogeneous semantic meanings of words are jointly modeled. To be more specific, $\boldsymbol{p}_i$ encodes the stationary distribution of random walks starting from node i, it can be interpreted as the relevant weights of other nodes (i.e., words) to the seed node (i.e., selected word) in this graph G (i.e., input news article). Our input node feature $\boldsymbol{x}_i$ is general because it is distilled by pre-trained models from a large-scale corpus like Wikipedia. Thus, according to Eq. 1, $\boldsymbol{x}_i$ is in-depth specialized by $\boldsymbol{h}_i$, which means the meaning of a selected word is specialized by its neighbor words in the scope of this input article.

Then, to obtain the graph-level (i.e., document-level) representation $\boldsymbol{u} \in \mathbb{R}^{1\times d}$ for the word graph G (i.e., input news article), we read out all word-level hidden representations $\boldsymbol{h}_i$ as follows

$$\boldsymbol{u} = readout(\boldsymbol{h}_i \mid i \in \{1, \ldots, n\}) \tag{3}$$

where the *readout* function is permutation-invariant and could be the graph pooling layer, such as sum or average pooling [21, 24].

This geometric feature extraction is not only effective because it replaces the traditional message-passing scheme of stacking GNN layers with the stationary distribution based aggregation [5, 12]; but is also efficient, which means the new stationary distribution can be fast tracked when the graph topology changes [8–11], e.g., in this demo, we mask several nodes in the word graph. These two merits pave the way for our explanation function of DISCO.

3.3 Neural Detection

The loss function of the proposed DISCO model is deployed on the representation vector $\boldsymbol{u}_j$ against the label information $\boldsymbol{y}_j$ (e.g., fake or real) of the j-th news article. To realize this, we need to call a neural network $\mathbb{N}$ to transform each $\boldsymbol{u}_j$ into $\boldsymbol{z}_j$. For instance, the cross-entropy between these two is expressed as follows

$$\mathcal{L} = -\sum_j \boldsymbol{y}_j \ln \boldsymbol{z}_j \tag{4}$$

where $\boldsymbol{z}_j \in \mathbb{R}^{1\times q}$ is the final output of neural network $\mathbb{N}$ w.r.t $\boldsymbol{u}_j$, and $\boldsymbol{y}_j \in \mathbb{R}^{1\times q}$ denotes the ground truth label of the entity j.

Benefiting from our geometric feature extraction scheme, the deployment of neural network $\mathbb{N}$ can be model-agnostic [5, 9, 12], which means the feature extraction is independent of the neural detection, and we can apply various kinds of neural networks according to different settings. For example, in our online demo, we instance $\mathbb{N}$ with a simple 32*2 multi-layer perceptron (MLP) for achieving effective performance, as shown in Figure 2. Additionally, we also provide the contextual multi-armed bandits in the exploitation-exploration dilemma [3, 4].

3.4 Explanation of Misleading Words

Given an article is detected as fake or real, each word has different contributions to this decision. For example, some words help disinformation camouflage and hider the detection to detect it. After removing such words in that article, the decision can be more deterministic, i.e., the corresponding prediction probability increases. In this paper, we call these words "misleading words". Next, we explain how our DISCO could explain each word's misleading degree.

We can choose to mask any nodes in the word graph G_j to see their contributions to the final representation $\boldsymbol{z}_j$, to further see to what extent the prediction is changed. Therefore, we can know what factors make DISCO dictate such a prediction, which is especially helpful for misclassified cases. Technically, masking nodes and edges without re-training the model from scratch relies on our proposed Eq. 1 and Eq. 3, where $\boldsymbol{p}_i$ is the stationary distribution of seed node i on graph G. When we need to mask a certain node in graph G and change it into G', the new stationary distribution $\boldsymbol{p}'_i$ can be fast and accurately tracked only with the topology changes but without the neural network parameters fine-tuning [9].

In Eq. 2, we use $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{D}^{-1} \in \mathbb{R}^{n\times n}$ to denote the column-stochastic transition matrix. When a certain node is masked (i.e., its adjacent edges are deleted), the graph topology will change from $\boldsymbol{M}$ to $\boldsymbol{M}'$, and the new stationary distribution $\boldsymbol{p}'_i$ of each node i needs to be tracked. To obtain each $\boldsymbol{p}'_i$, the core idea is to push

out the previous probability distribution score from the changed part to the residual part of the graph G, and then add the pushed out distribution $\boldsymbol{p}_i^{pushout}$ back to the previous distribution $\boldsymbol{p}_i$ to finally obtain the new distribution $\boldsymbol{p}_i'$. The tracking process can be described as follows

$$\boldsymbol{p}_i^{pushout} = \alpha(\boldsymbol{M}' - \boldsymbol{M})\boldsymbol{p}_i \tag{5}$$

and

$$\boldsymbol{p}_i' = \boldsymbol{p}_i + \sum_{k=0}^{\infty}(\alpha\boldsymbol{M}')^k\ \boldsymbol{p}_i^{pushout} \tag{6}$$

where $\boldsymbol{p}_i^{pushout}$ denotes the distribution score that needs to be pushed out on the residual graph due to the updated edges, and $\boldsymbol{p}_i'$ denotes the tracked new distribution. The above pushout process can be proved to converge to the exact stationary distribution of the new graph through sufficient cumulative power iterations [22, 23].

After we get each new $\boldsymbol{p}_i'$, according to Eq. 1 and Eq. 3, we can get the new graph-level hidden representation $\boldsymbol{u}'$ and final representation $\boldsymbol{z}'$ without fine-tuning neural network $\mathbb{N}$. Then the difference between the correct prediction probability of $\boldsymbol{z}'$ and $\boldsymbol{z}$ composes the misleading degree of the masked word. For example, as shown in Figure 2, when we mask the word "recreational" in the input news article, the new probability of predicting this new article (i.e., $\boldsymbol{z}'$) as fake news is 99.999999415%. Compared with 99.999998466% of the original article (i.e., $\boldsymbol{z}$), the confidence of correct prediction increases 0.000000949% by masking the word "recreational". Therefore, the word "recreational" hinders DISCO make the correct prediction, and its misleading degree is 0.00000000949. In our demo, we rank each word based on its misleading degree and show the rank in decreasing order. Also, the misleading degree can be negative, which means that the masked word in the original article helps DISCO make correct predictions.

4 REAL-WORLD EVALUATION

4.1 Experiment Preparation

Data Set. We choose the fake real news data set from [1, 2], and each news article focusing on politics around the world has a title, text, subject, date, and a label indicating it is fake news or not. After preprocessing, the statistics of valid records of fake news and real news are stated as below. The fake news articles consist of 23,481 items, ranging from Mar 31, 2015 to Feb 19, 2018. As for real news articles, 21,417 items are involved from Jan 13, 2016 to Dec 31, 2017.

Environment Setup. First, we initialize Google pre-trained Word2Vec [4] for the node input feature $\boldsymbol{X}$ in Eq. 1. Second, we use the sum-pooling function as the *readout* function in Eq. 3. Third, we use a multi-layer perceptron with the ReLU activation function to distill $\boldsymbol{u}$ in Eq. 4, where the optimizer is based on Adam and the initial learning rate is set to be 0.0001. The experiments are programmed based on Python 3.7 in a Windows machine with four 3.6GHz Intel Cores and 64GB RAM.

4.2 Performance of DISCO

Accuracy. We show the performance of different baselines with our DISCO and report their fake news detection accuracy in Figure 3. For each bar in Figure 3, we report the performance w.r.t the average and standard deviation under the 80%–20% training-testing split 10 times randomly. From Figure 3, we can see that DISCO achieves the best performance. A possible interpretation is that the graph modeling of DISCO is better to model the relationship between words in articles than sequential modeling (i.e., LSTM) or non structural modeling (TF-IDF + KNN), $k = 3$.

Robustness. From the ablation study as shown in Figure 4, we can observe that (1) even the sample neural detection module can achieve competitive performance under the DISCO modeling, for MLP (hidden dim = 32, # layers = 2) achieving the most accurate and stable predictions; (2) in many cases, too few or too many training samples could not contribute the most to the neural detection module under our DISCO modeling, for MLP (hidden dim = 64, # layers = 1) and MLP (hidden dim = 64, # layers = 2) reaching the peak when 20% of the whole data set are extracted as testing samples.

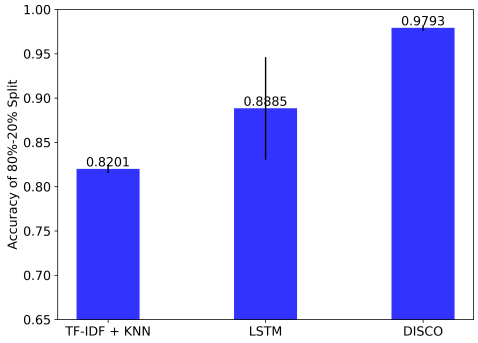

Figure 3: Performance on the 80%-20% Split Setting.

Sensitivity. From Figure 4, we know that cooperating with MLP (hidden dim = 32, # layers = 2) our DISCO is robust for the low variance with varying testing sample sizes. Here, we want to investigate the prediction sensitivity of DISCO. Again, based on the 75%–25% training-testing split, we shuffle the entire data set 10 times randomly and report the precision, recall, and F1-score. The precision of DISCO is 0.9748 ± 0.0086, which means that in all the positive predictions, 97.48% of them are correct predictions. The recall of DISCO is 0.9754 ± 0.0065, which means that among all ground-truth positive items, 97.54% of them are identified by DISCO. The average and standard deviation of the F1-score is 0.9750 ± 0.0010, and the low standard deviation suggests that each time we achieve high precision, we also achieve a high recall simultaneously.

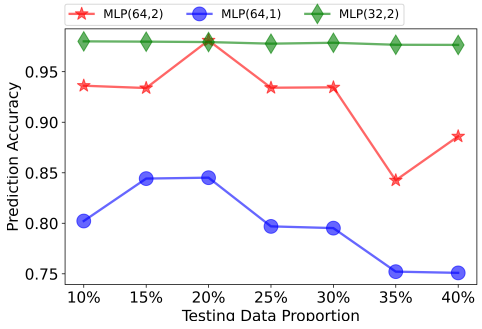

Figure 4: Performance of DISCO on Different Settings.

5 CONCLUSION

In this paper, we identify disinformation detection challenges and make an attempt for proposing DISCO and demonstrate it. We wish that the next generation of disinformation detection systems could be able to simultaneously detect and explain during the whole life cycle of disinformation dissemination.

[4] https://code.google.com/archive/p/word2vec/

REFERENCES

[1] Hadeer Ahmed, Issa Traoré, and Sherif Saad. 2017. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In *ISDDC 2017*.

[2] Hadeer Ahmed, Issa Traoré, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Secur. Priv.* (2018).

[3] Yikun Ban and Jingrui He. 2021. Convolutional neural bandit: Provable algorithm for visual-aware advertising. *arXiv preprint arXiv:2107.07438* (2021).

[4] Yikun Ban, Jingrui He, and Curtiss B. Cook. 2021. Multi-facet Contextual Bandits: A Neural Network Perspective. In *KDD 2021*.

[5] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. 2020. Scaling Graph Neural Networks with Approximate PageRank. In *KDD 2020*.

[6] Bjarte Botnevik, Eirik Sakariassen, and Vinay Setty. 2020. BRENDA: Browser Extension for Fake News Detection. In *SIGIR 2020*.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019*.

[8] Dongqi Fu and Jingrui He. 2021. DPPIN: A Biological Repository of Dynamic Protein-Protein Interaction Network Data. *CoRR* (2021).

[9] Dongqi Fu and Jingrui He. 2021. SDG: A Simplified and Dynamic Graph Neural Network. In *SIGIR 2021*.

[10] Dongqi Fu, Zhe Xu, Bo Li, Hanghang Tong, and Jingrui He. 2020. A View-Adversarial Framework for Multi-View Network Embedding. In *CIKM 2020*.

[11] Dongqi Fu, Dawei Zhou, and Jingrui He. 2020. Local Motif Clustering on Time-Evolving Graphs. In *KDD 2020*.

[12] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR 2019*.

[13] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *EMNLP 2004*.

[14] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS 2013*.

[15] Sebastião Miranda, David Nogueira, Afonso Mendes, Andreas Vlachos, Andrew Secker, Rebecca Garrett, Jeff Mitchell, and Zita Marinho. 2019. Automated Fact Checking in the News Room. In *WWW 2019*.

[16] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2018. CredEye: A Credibility Lens for Analyzing and Explaining Misinformation. In *WWW 2018*.

[17] Prathusha Kameswara Sarma, Yingyu Liang, and Bill Sethares. 2018. Domain Adapted Word Embeddings for Improved Sentiment Classification. In *ACL 2018*.

[18] Prathusha Kameswara Sarma, Yingyu Liang, and William A. Sethares. 2019. Shallow Domain Adaptive Embeddings for Sentiment Analysis. In *EMNLP 2019*.

[19] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. dEFEND: Explainable Fake News Detection. In *KDD 2019*.

[20] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *SIGKDD Explor.* (2017).

[21] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *NeurIPS 2018*.

[22] Minji Yoon, Woojeong Jin, and U Kang. 2018. Fast and Accurate Random Walk with Restart on Dynamic Graphs with Guarantees. In *WWW 2018*.

[23] Minji Yoon, Jinhong Jung, and U Kang. 2018. TPA: Fast, Scalable, and Accurate Method for Approximate Random Walk with Restart on Billion Scale Graphs. In *ICDE 2018*.

[24] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In *AAAI 2018*.

[25] Zijian Zhang, Koustav Rudra, and Avishek Anand. 2021. FaxPlainAC: A Fact-Checking Tool Based on EXPLAINable Models with HumAn Correction in the Loop. In *CIKM 2021*.

[26] Shi Zhi, Yicheng Sun, Jiayi Liu, Chao Zhang, and Jiawei Han. 2017. ClaimVerif: A Real-time Claim Verification System Using the Web and Fact Databases. In *CIKM 2017*.