Semi-Supervised Learning and Graph Neural Networks for Fake News Detection

Adrien Benamira*, Benjamin Devillers*, Etienne Lesot*, Ayush K. Ray*, Manal Saadi*, Fragkiskos D. Malliaros*†
*CentraleSupélec, University of Paris-Saclay, France
Email: {adrien.benamira, benjamin.devillers, etienne.lesot, manal.saadi}@supelec.fr, ayush.rai2512@student-cs.fr
†Inria Saclay, France
Email: fragkiskos.malliaros@centralesupelec.fr

***Abstract*—Social networks have become the main platforms for information dissemination. Nevertheless, due to the increasing number of users, social media platforms tend to be highly vulnerable to the propagation of disinformation – making the detection of fake news a challenging task. In this work, we focus on content-based methods for detecting fake news – casting the problem to a binary text classification one (an article corresponds to either fake news or not). The main challenge here stems from the fact that the number of labeled data is limited; very few articles can be examined and annotated as fake. To this extend, we opted for semi-supervised learning approaches. In particular, our work proposes a graph-based semi-supervised fake news detection method, based on graph neural networks. The experimental results indicate that the proposed methodology achieves better performance compared to traditional classification techniques, especially when trained on limited number of labeled articles.**

***Index Terms*—Fake news detection, Semi-supervised learning, Graph neural networks**

I. Introduction

Social media have become the main platforms for information sharing and news consumption for various reasons. Firstly, it is often a faster and cheaper way to access news on social media compared to more traditional platforms. Furthermore, commenting, sharing and discussing with other readers is an easy way to express opinions and increases the level of participation and interaction of individuals. Nevertheless, the ease with which real-time information disseminates to a large audience accompanied with the engagement of individuals to online social media platforms, has also lead to the spread of misinformation, widely known as *fake news* [12]. Fake news take advantage of the *echo chambers* phenomenon, amplified by social networks; people tend to follow and share mainly information they believe in or what their friends share and like. This is why social media platforms are particularly vulnerable to the propagation of fake news, mostly coming from unverified publishers and crowd-based content creators.

Triggered by the societal impact of misinformation, there is currently an intense research effort from the scientific community to develop algorithmic techniques for fake news detection. The core of these techniques relies on machine learning methods – trying to analyze and understand how the content of fake news differs from that of real ones, as well as how users engage with and propagate misinformation within social networks [13], [14].

A large number of works rely on handcrafted features extracted from news content and supervised classification models [11]. Hoerne and Adali [15] identified various text characteristics that differentiate the content of fakes news from real ones. Their main findings indicate that fake news articles follow a set of specific writing rules: longer titles, more capitalized words, fewer stop words and automatic shorter body. In a more recent study, Pérez-Rosas [16] observed that fake news articles contained more temporal words, indicating that the content of the article tends to be focused on the present and future. Nevertheless, linguistic analysis methods require hand-crafted feature extraction and cannot model more complex contextual dependencies alone. Deep learning methods alleviate the shortcomings of linguistic methods by automatic feature extraction, demonstrating significant performance in text classification. Wang [17] proposed to use convolutional neural networks (CNNs) for content-based fake news detection. Other approaches include sentence and word level CNNs [18] and long short-term memory (LSTM) networks [19]. Nevertheless, as a recent evaluation study has indicated [20], the content-based classification accuracy achieved with CNNs is relatively low. Moreover, such approaches typically require a significant amount of labeled data which, in many cases, is hard to obtain.

In order to tackle the fact that labels are often very limited and sparse, we opt here for *semi-supervised* content-based detection methods. In particular, we propose a graph-based semi-supervised fake news detection framework, building upon network representation learning techniques [21]. Our intuition is that, graphs are expressive models that are able to capture contextual dependencies among articles, alleviating the label scarcity constraint [1]. On a high level, our framework is composed of three components: i) embedding of articles in the Euclidean space; ii) construction of an article similarity graph; iii) inference of missing labels using graph learning techniques. The main contributions of this paper are summarized as follow:

- We use word embeddings to obtain latent representations of news articles in a lower dimensional Euclidean space. Then, we capture contextual similarities among articles via a graph-based representation scheme.

- We cast the fake news detection problem as a semi-supervised graph learning task, leveraging Graph Neural Network architectures that are able to perform well on limited labeled data.
- We perform a preliminary evaluation of our methodology on a real fake news dataset, demonstrating that the proposed methodology outperforms previous content-based approaches, requiring fewer labeled articles.

II. Background Concepts: Graph Neural Networks and Extensions

A. Graph Neural Networks (GNNs)

Initially, GNNs were introduced as an extension of Recurrent Neural Networks (RNNs); GNNs apply recurrent layers to each node with additional local averaging layers [3], [4]. The goal of GNNs is, given a feature vector X and a graph A, to find a model f predicting at each node one of d_y label classes $f(X, A) = Z \in \mathbb{R}^{I \times d_y}$, where Z_{ic} is the estimated probability that the label at node $i \in \{1, \cdots, n\}$ is $c \in \{1, \cdots, d_y\}$.

B. Graph Convolutional Network (GCN)

Graph Convolutional Network (GCN) [5] is a special case of GNNs which stacks two layers of specific propagation and perceptron:

$$\begin{aligned} H^{(1)} &= \text{ReLU}\left((PX)W^{(0)}\right) \\ Z = f(X, A) &= \text{Softmax}\left(PH^{(1)}W^{(1)}\right) \end{aligned} \tag{1}$$

with a choice of $P = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$, where $\tilde{A} = A+I$, where I is the identity matrix, $\tilde{D} = \text{diag}(\tilde{A}\vec{1})$ and $\vec{1}$ is the all ones vector. The weights are trained to minimize the cross-entropy loss over all labeled examples L:

$$\mathcal{L} = -\sum_{i \in L}\sum_{c=1}^{d_y} Y_{ic}\log(Z_{ic}) \tag{2}$$

The problem with GCN is the fact that they are weight-consuming which is critical for semi-supervised learning where the number of labeled examples is small. Besides, there is a lack of interpretability.

C. Attention-based Graph Neural Network (AGNN)

The attention-based Graph Neural Network (AGNN) [2] corresponds to a novel graph neural network architecture that removes all the intermediate fully-connected layers, and replaces the propagation layers with an attention mechanism which respects the structure of the graph. The attention mechanism allows to learn a dynamic and adaptive local summary of the neighbourhood, achieving more accurate predictions. In addition, the attention-based graph neural network is able to:

- Greatly reduce the model complexity, with only a single scalar parameter at each intermediate layer;
- Discover dynamically and adaptively which nodes are relevant to the target node for classification.

III. Proposed Approach

A schematic representation of our approach is depicted in Figure 1. In order to use graph-based algorithms for fake news detection, we first construct a graph out of the dataset, following similar methodological ideas as in [1]. Our approach is based on the following steps: document embedding and graph inference for the representation of articles, and graph neural network architectures for classification[1].

A. Embedding of Articles and Graph Inference

Let $\mathcal{N} = \{n_1, n_2, n_3, \ldots, n_M\}$ be a collection of M articles, where each article is a vector that contains the words within the article. The graph $\mathcal{G} = (V, E)$ will be designed such that the set of nodes V will contain the articles and we will consider that an edge $(v_1, v_2) \in E$ connects two articles if the two articles are *close* enough in the embeddings space. We have also removed the most common words for the documents.

1) Embedding of articles: To define the notion of *distance*, we will embed an article into a vector space and use a simple Euclidean distance.

- Co-occurence matrix and CP/PARAFAC tensor decomposition [1]: we build a three-mode tensor $X \in \mathbb{R}^{I \times I \times M}$ (words, words, articles), where all co-occurrence entries are boolean and indicate whether the i^{th} and j^{th} words appear within a predefined window at least once. We then use CP/PARAFAC tensor decomposition to factorize the tensors. A tensor is a multi-dimensional array, where each dimension represents a mode. Canonical Polyadic (CP) or PARAFAC decomposition is a tensor decomposition method, widely used that factorizes a tensor into a sum of rank one tensors.
- Pre-trained GloVe word embedding: this simple method consists of considering the mean of the pre-trained GloVe word embeddings for each word [7], [8] to get the embedding of one article. GloVe is a model that learns geometrical encodings (vectors) of words from their co-occurrence information (i.e., how frequently they co-occur in a large text corpora). GloVe is a *count-based model*, meaning that it learns vector representations by doing dimensionality reduction of the co-occurrence count matrix. To obtain the vector representation of the article from the representations of the words, we have used the averaging the vectors.
- Latent Dirichlet Allocation (LDA): LDA is a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document [10].

[1]Our code is available here: https://github.com/bdvllrs/misinformation-detection-tensor-embeddings.

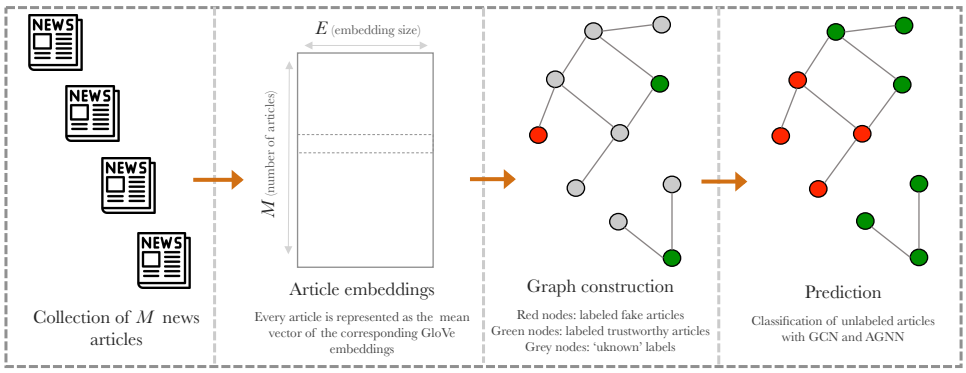

Fig. 1. Illustration of the proposed approach: M denotes the number of articles (real and fake) and E is the dimension of our GloVe embeddings (in our case, $M = 150$, $E = 100$). Finally, we use $k = 4$ nearest neighbours to build the graph.

2) k–nearest-neighbours graph construction: After transforming the article into a vector, we can then create our graph. For each node (article) we look for the k–nearest-neighbours (k–nearest articles) using a simple Euclidean distance in the embedding space. We havel also forced the graph to be undirected. Finally, as we will present shortly, we tuned the number of neighbours k and it appears that we obtain similar results with values ranging from 1 to 10.

B. Classification

For the classification task over the similarity graph between articles, we use Graph Convolutional Networks (GCN) [5] and Attention Graph Neural Networks [2], as presented in Sec. II.

IV. Experimental Evaluation

A. Experimental Set-up

In our empirical evaluation, we have compared our graph neural network-based methods against the approach by Guacho et al. [1], which follows a similar framework. In that method, the embedding of articles is obtained using CP/PARAFAC tensor decomposition on the binary co-occurrence matrix between all articles, and the classification is performed using the Fast Belief Propagation (FaBP) algorithm [9]. We have also compared against a traditional approach, which consists of bag-of-words textual features and learning with SVMs and Random Forest classification models. For the neural graph networks, we use 4 layers, 4 neighbours, 16 hidden units, a learning rate of 0.01 and a weight decay of $5e-4$. We train our graph neural networks during 1000 epochs and we keep the one which has the best accuracy on the test (unlabeled) data.

We evaluate our method on a recent dataset [11], which is comprised of 150 labeled articles, 75 of those are fake news and 75 real. We pick the labeled articles at random and average the results over 20 independent experiments. We force our training subset to be balanced. That is to say, we force the labeled subset to have half of fake news data and half of real news. We also constructed the graph for $k = 1, 2, 3, 3, 4$ neighbors and study the influence of k.

B. Results

Table I gives the classification accuracy of the different methods, varying the amount of labeled data used for training (ranging from 2% up to 20%). As we can observe, the proposed graph neural network approaches achieve a performance improvement of up to 3% with only 10% of the labeled data, while being more stable and reducing the standard deviation of the results. Furthermore, our AGNN and GCN methods are computationally faster when it comes to evaluate a new article. Next, we highlight some key observations from the experimental evaluation.

1) Influence of the embeddings: In Figure 2, we compare the performance of the various embedding methods. As we can observe, the GloVe embedding approach yields the best results. Moreover, the co-occurence embeddings perform quite similar to GloVe embeddings, especially for large fractions of labeled data.

2) Influence of the number of neighbours: In Table II, we examine the impact of the number of neighbors k used to build the graph. Contrary to the approach by Guacho et al. [1] where the number of neighbours heavily impacts the quality of the results, no such deviation can be observed when the classification is performed using the convolutional-based graph neural networks (GCN). This can be explained by the fact that we build the k-nearest-neighbour graph with the same information that we give to the network for classification.

V. Conclusion and Further Work

In this paper, we have focused on content-based misinformation detection, aiming at classifying fake news relying solely on the content of the article — while assuming that we have a limited amount of labeled articles. The preliminary experimental results have suggested that building a simple nearest-neighbour graph among articles based on word embedding similarities accompanied by graph neural networks for classification, give qualitatively good results — providing the basis for semi-supervised content-based detection methods. We are currently working to further extend our study by considering more baseline methods and testing the performance on bigger as well as multi-labeled fake news datasets.

Methods	**Accuracy (in %)**				
	2 % labeled data	*5 % labeled data*	*10 % labeled data*	*15 % labeled data*	20 % labeled data
Guacho et al. [1]	56.65 ± 9.67	63.60 ± 7.52	70.95 ± 5.28	74.05 ± 3.80	79.8 ± 3.10
SVM	63.55 ± 5.73	66.55 ± 7.14	75.05 ± 5.20	76.05 ± 4.80	78.90 ± 5.16
Random Forest	60.25 ± 10.02	69.05 ± 3.33	76.65 ± 3.48	**83.55 ± 5.06**	84.70 ± 2.48
AGNN	70.45± 5.39	72.00 ± 8.05	78.70 ±3.54	83.35 ± 1.74	84.25 ± 3.51
GCN	**72.04 ± 6.00**	**77.35 ± 3.72**	**79.85 ± 3.41**	82.35 ± 2.44	**84.94 ± 2.30**

TABLE I
CLASSIFICATION ACCURACY WITH DIFFERENT RATIOS OF LABELED DATA USED FOR TRAINING WITH $k = 4$.

k (number of neighbors)	**Accuracy (in %)**				
	2 % labeled data	*5 % labeled data*	*10 % labeled data*	*15 % labeled data*	20 % labeled data
$k = 1$	**74.64 ± 5.02**	**79.25 ± 3.65**	82.7 ± 2.82	**85.00 ± 2.10**	**86.95 ± 2.82**
$k = 2$	69.00 ± 7.39	76.85 ± 4.47	**82.95 ± 4.03**	83.40 ± 3.23	83.94 ± 3.63
$k = 3$	71.80 ± 8.58	74.90 ± 5.09	78.95 ± 3.45	82.95 ± 3.42	83.75 ± 5.13
$k = 4$	72.04 ± 6.00	77.35 ± 3.72	79.85 ± 3.41	82.35 ± 2.44	84.94 ± 2.30

TABLE II
ACCURACY WITH k NEIGHBORS AND GCN CLASSIFICATION.

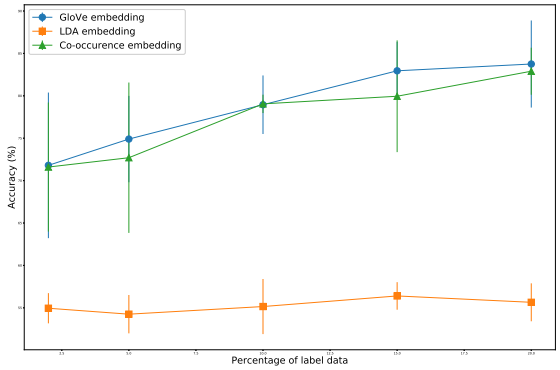

Fig. 2. Impact of embedding method. Comparison of the classification accuracy based on GCN for various embedding methods, with respect to the percentage of labeled articles. Four neighbours are used to build the graph.

REFERENCES

[1] Guacho, G. B., Abdali, S., Shah, N., Papalexakis, E. E. (2018, August). 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.

[2] Thekumparampil, K. K., Wang, C., Oh, S., Li, L. J. (2018). Attention-based graph neural network for semi-supervised learning. arXiv preprint arXiv:1803.03735.

[3] Gori, M., Monfardini, G., Scarselli, F. (2005, July). A new model for learning in graph domains. In Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. (Vol. 2, pp. 729-734). IEEE.

[4] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., Monfardini, G. (2009). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61-80.

[5] Kipf, T. N., Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[6] Yao, L., Mao, C., Luo, Y. (2018). Graph convolutional networks for text classification. arXiv preprint arXiv:1809.05679.

[7] Pennington, J., Socher, R., Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[8] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[9] Koutra, D., Ke, T. Y., Kang, U., Chau, D. H. P., Pao, H. K. K., Faloutsos, C. (2011, September). Unifying guilt-by-association approaches: Theorems and fast algorithms. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 245-260). Springer, Berlin, Heidelberg.

[10] Blei, D. M., Ng, A. Y., Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

[11] Horne, B. D., Adali, S. (2017). This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In Eleventh International AAAI Conference on Web and Social Media.

[12] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. (2017). Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

[13] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu. (2019). Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology*, 2019.

[14] X. Zhou and R. Zafarani. (2019). Fake news: Fundamental theories, detection strategies and challenges. *arXiv*.

[15] B. D. Horne and S. Adali. (2017). This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *ICWSM*.

[16] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea. (2018). Automatic detection of fake news. In *ACL*, pp. 3391–3401.

[17] W. Y. Wang. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *ACL*.

[18] F. Qian, C. Gong, K. Sharma, and Y. Liu. (2018). Neural user response generator: Fake news detection with collective user intelligence. In *IJCAI*, pp. 3834–3840.

[19] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *EMNLP*, pp. 2931–2937.

[20] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. (2018). FakeNewsNet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv*.

[21] W. L. Hamilton, R. Ying, and J. Leskovec. (2017). Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74.