On The Pursuit of Fake News : Graph Neural Network meets NLP

Zeynep Pehlivan
Institut national de l'audiovisuel, France
zpehlivan@ina.fr

ABSTRACT

This paper presents the methods proposed by FakeINA team to participate The FakeNews: Corona Virus and Conspiracies Multimedia Analysis tasks. We concentrate our work on text-based misinformation and conspiracy detection. We proposed a multimodal neural network that combines a graph neural network (GNN) where a document is represented as graph and a multi-layer perceptron model where textual statistics are used as features. Experimental results show that however GNNs are able to classify the data, a multimodal performs better.

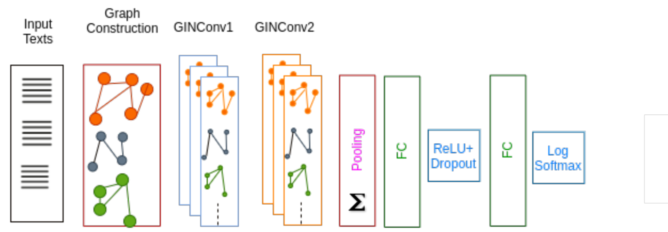

Figure 1: Our framework of GIN with 2 layers for graph classification

1 INTRODUCTION

Mediaeval Fake News task[6, 7] focuses on the classification of tweet texts aiming detection of fast spreading misinformation. This task contains three sub-tasks : Text-Based Misinformation Detection, Text-Based Conspiracy Theories Recognition and Text-Based Combined Misinformation and Conspiracies Detection. This work proposed a multimodal neural network approach which is only applied to the first two sub-tasks.

Graph neural network (GNN) methods have been profoundly useful in several domains including natural language processing[9]. While it is probably most apparent to regard text as sequential data, there are several methods to represent text as various kinds of graphs. Dependency graph construction generates a graph by extracting the dependency relations from the dependency parsing tree. Constituency graph construction captures phrase-based syntactic relations in a sentence. Another way of representing text as a graph is to use word co-occurrence and/or document word relations.

TextGCN [11] proposes to build a single heterogeneous graph for whole corpus and captures global word co-occurrence information. This approache converts text classification task to node classification task. On the other hand, TextING[12] builds a graph for each document based on the co-occurrence of words where each node is represented as a word embedding and sliding window is used to capture the relation between words. It learns the fine-grained word representation of the local structure by GNN to effectively produce embeddings for obscure words in the new text. By representing each document as a graph, text classification task becomes graph classification task for GNNs.

We present our approach in Section 2 and we discuss the results in Section 3.

MediaEval'21, December 13-15 2021, Online

2 APPROACH

In this section, we present the preprocessing applied to the raw text before graph construction. Two models proposed and their implementation details are also presented in this section.

2.1 Preprocessing

We use the spaCy[4] library for Python in order to create a text file containing the original tokens and their normalized counterparts. For this normalization we have transformed each token to lowercase and removed stop words. Also, considering the fact that BERT allows a maximum of 512 tokens per sequence and the given dataset contains sentences above that range, Bert tokenizer is used with truncation option. However, BERT is recommended to use with the raw text, lemmatization and stemming remain important to generate a graph since a node (word) would be disrupted by an irrelevant inflection like a simple plural.

2.2 Graph Construction

We use the same graph construction approach as described in TextING[12]. Each document is represented as a undirected graph where nodes are words and co-occurrences between words represented as edges. The co-occurrences is calculated by using a sliding window. Embedding of the nodes are initialized by extracting word embeddings from BERT model[2].

2.3 Models

After graph construction, the task converts to the graph classification task. The main idea behind GNNs is to compute a state for each node and update this state according to neighbouring nodes states at each iteration. Graph Isomorphism Network (GIN)[10] was proposed as a special case of spatial GNN suitable for graph classification tasks to overcome the issue of distinguishing non-isomorphic graphs. The authors argues that GIN is possibly as powerful as the Weisfeiler-Leman test [8] test for graph classification tasks. Thus, GIN is used in our experiments as GNN choice. Figure 1 resumes

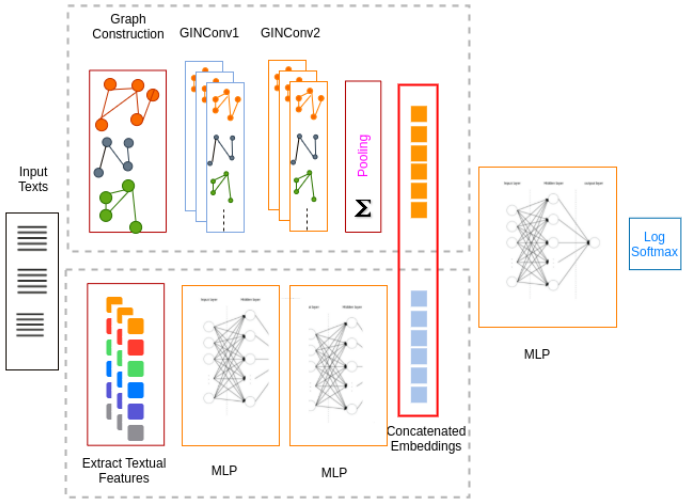

Figure 2: Our framework of multimodal approach with GIN and MLP

our framework where two GIN convolutional layers are followed by pooling layer (sum is used) and fully connected layers.

We also implemented a multimodal approach by combining GIN model with multilayer perceptron (MLP). As seen in Figure 2, we do not only generate graphs from input texts but also extract textual features s listed in Table 1, by using textstat[1] Python package. These features become input layer for MLP. We extract the embeddings from the last hidden layer and concatenat them with the graph embeddings obtained just after pooling layer. They are sent to MLP whose output layer will return the predictions for classification task.

flesch reading ease	syllable count
flesch kincaid grade	lexicon count
automated readability index	sentence count
dale chall readability score	char count
reading time	letter count
monosyllab count	emoji count

Table 1: Textual Features

2.4 Implementation

All the models are implemented by using Pytorh Geometric[3]. For text-based misinformation task, we used the negative log likelihood loss function, Adam optimizer [5], and StepLR scheduler. For the conspiracy detection we also used Adam optimizer [5] StepLR scheduler with binary cross entropy with logits loss function. As the dataset is not balanced, weights are provided for both loss functions. We implemented a grid search to find best values for sliding window size, number of GNN layer and hidden layers. Best value for sliding window size was 3 and number of GNN layer was 2.

3 RESULTS AND DISCUSSIONS

Stratified K-Fold cross validation model (with k=10) is used to measure the performance. For each fold, dataset is split into training(80%), validation (20%). Due to the small size of the dataset and overfitting issues during training we did not use test split. Table 2 shows the results for K-Fold CV by using Matthews correlation coefficient.

Task	Model	Hidden Layers	Val MCC	Official MCC
Task 1	multimodal	128	0.422	0.336
Task 1	multimodal	256	0.396	**0.446**
Task 1	GIN	256	0.390	0.384
Task 2	multimodal	64	0.325	0.223
Task 2	multimodal	128	0.331	**0.276**
Task 2	multimodal	32	0.325	0.21

Table 2: Stratified K-Fold CV and submission results

We observe that multimodal behaves better than GIN based approach for Task 1. For Task 2, we did not send any submission for GIN model because during experiments best Val MCC we get was 0.20. Hence, incorporating more data might improve the classification results.

REFERENCES

[1] 2018. textstat: Textstat is an easy to use library to calculate statistics from text. It helps determine readability, complexity, and grade level. (2018).

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2019). arXiv:cs.CL/1810.04805

[3] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[4] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017).

[5] Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[6] Konstantin Pogorelov, Daniel Thilo Schroeder, Stefan Brenner, and Johannes Langguth. 2021. FakeNews: Corona Virus and Conspiracies Multimedia Analysis Task at MediaEval 2021. *Proc. of the MediaEval 2021 Workshop* (2021).

[7] Konstantin Pogorelov, Daniel Thilo Schroeder, Petra Filkuková, Stefan Brenner, and Johannes Langguth. 2021. WICO Text: A Labeled Dataset of Conspiracy Theory and 5G-Corona Misinformation Tweets. *Proc. of the 2021 Workshop on Open Challenges in Online Social Networks* (2021).

[8] Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series* 2, 9 (1968), 12–16.

[9] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2021. Graph Neural Networks for Natural Language Processing: A Survey. *arXiv:2106.06090 [cs]* (June 2021). http://arxiv.org/abs/2106.06090 arXiv: 2106.06090.

[10] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference*

on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net. https://openreview.net/forum?id=ryGs6iA5Km
[11] Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph Convolutional Networks for Text Classification. *arXiv:1809.05679 [cs]* (Nov. 2018). http://arxiv.org/abs/1809.05679 arXiv: 1809.05679.
[12] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. *arXiv:2004.13826 [cs]* (May 2020). http://arxiv.org/abs/2004.13826 arXiv: 2004.13826.