A Framework for Predicting Sources of Malicious Information on Social Media Networks

Sameer Gupta
Department of Computer Science
Columbia University
New York, NY
sg4021@columbia.edu

Mahesh Jindal
Department of Computer Science
Columbia University
New York, NY
mj3038@columbia.edu

Abstract

Malicious information in the form of misinformation and hate speech has proliferated social networks. Current approaches are prone to adversarial attacks and fail to capture the effect of the propagation path and the user interactions. In this paper, we propose a framework based on the existing UPFD (6) approach. We augment the existing model with (i) document embeddings of the root source, (ii) a weighted textual representation of the users old tweets based on the proportion of malicious in their history, and (iii) controlling the proportion of structural signal and the textual signal. We extend our approach to detecting hate speech as well as test our approach on newer data. Our results show that this approach can be extended to related malicious information tasks.

1 Introduction

The excess proliferation of misinformation on social media platforms is a poignant reminder of the inadequacy of generic automated large scale misinformation detection systems (11). This influx has motivated a large proportion of current research to build scalable systems to tackle this problem (4; 7). The *bursty* nature of misinformation is another notion of *virality* on social media networks, which is their inherent characteristic (15). Virality on social networks thus makes it hard for traditional machine learning systems to detect misinformation. This is due to the ever evolving nature of viral news, which in turn is heavily dependant on worldy events. Though there are systems in place to tackle such issues (5), these systems are prone to adversarial effects (1), struggle to take into account the user context and ultimately fail to *contain* the source of the misinformation. In this paper, we propose a architectural revamp of the famous UPFD (6) framework. This framework stems from the idea that we must focus on the graph propagation structure of a piece of news as well as the social context of the nodes who are involved in this propagation. Our proposition is that we can extend this architecture to generalize for any kind of misinformation, be it an external article or an internal post, by augmenting it with additional modules. We also extend this architecture to hate speech. As at its core, our system is a supervised learning system, generalization is of the utmost importance.

The general transmission mechanism of fake news in terms of its life-cycle was aptly captured by Zhou et. al. (17). Most machine learning based systems work on classifying misinformation once the propagation step is in full force. Further, they only focus on the isolated textual/image content of the information rather than a contextual understanding of propagation. The general pipeline to detect misinformation consists of first using a large corpora of labelled data (fake vs real) to query the text and check if there are similar documents related to that text (information retrieval). Next step is detecting the stance of the text using natural language inference models. This tells us whether the retrieved information entails, contradicts, or is neutral w.r.t. the text. This pipeline, combined with a set of supervised learning models to weakly label the data is usually the industry standard.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

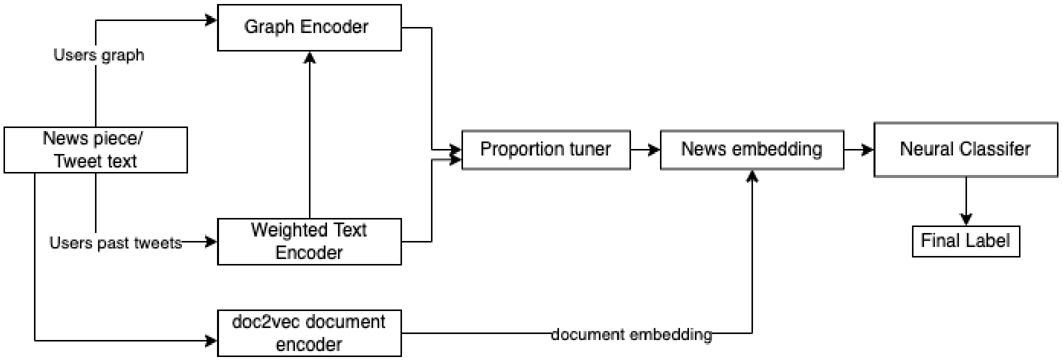

Figure 1: Our modified architecture based on the UPFD approach

With the advancement in sequence networks and language modelling, they are actively used for this classification task (8; 10). Islam et. al. (8) encapsulates a variety of approaches which use both recurrent discriminator based approaches using sequence models such as LSTMs, RNNs and also generative models such as GANs. More recently, many researchers have began to realise the importance of social features of nodes in the network. These features encode the position of node in the network. This position is a function of closeness between nodes, centrality etc. Graph based deep learning models have been gaining traction for tasks such as node classification, link prediction (16) etc. A particular type of Graph Neural Network (GNN) is the graph auto encoder (GAE). Graph convolutional networks are types of GAEs which essentially map a node's representation into another latent representation. The property of GCNs which is of interest to us is its ability to not only look at the node $v \in G$ but also the set of neighbors $u \in N(v)$. This means that the embedding E_v will be a function of not only the features and position of the node v but also its neighbors (16). This property will be of use to us to encode social graph based context.

In our approach, we expand on the seminal UPFD approach (6). The approach by Dou et. al. (6) is based on the seminal dataset by Shu et. al. (13). The task is to detect if a news article is misinformation or not using it's social interaction on a social media network. Particularly, their approach focused on using the positional node encoding of the users from the retweet tree/graph of particular news articles along with textual embeddings of the users content. We extend this approach by proposing the following:

- Utilizing more recent data examples dealing with datasets ranging from the 2016 US Presidential Elections, the Covid-19 pandemic discourse and the 2020 BLM movement. We also introduce the task of detecting hate speech using these datasets.
- Adding a document encoder as an additional signal for misinformation classification.
- Augmenting the classic GCN with models such as Graph Attention Networks (GAT) (14) and Hypergraph Convolution Networks (2).
- We use the past tweets of the user which captures the propensity of a user to retweet misinformation/hate speech
- Adding a parameter that controls the proportion of the information of structural GNN information and the textual contextual user information.

2 Our Approach

In this section, we elaborate on the different components that build on top of the UPFD (6) approach. As can be seen in Figure 1 , our model greatly enhances the base UPFD model.

2.1 Collection of temporally relevant data

We collect Twitter data to create similar tree structured graph and assess the performance of our model. For the 2016 US election data, we borrow data from Ribeiro et. al. (12). This data has 200 most recent tweets of 100,386 users which accounts to about 19M tweets A retweet induced graph with

Dataset	Trees	Nodes	Edges
$Dataset_{base}$ POL	314	41,054	40,740
$Dataset_{base}$ GOS	5464	314,262	308,798
$Dataset_{hate}$	1	30,294	1,32,855
$Dataset_{covid}$	117	21,998	47,084

Table 1: Description of our datasets

2,286,592 directed edges is also provided. The retweet-induced graph is a directed graph $G = (V, E)$ where each node $u \in V$ represents a user in Twitter, and each edge $(u_1, u_2) \in E$ represents a user u_1 retweeting user u_2. We use this dataset particularly for hate speech. Out of the 100,386 users, 544 have been labelled hateful so we create an induced subgraph using these users. Then, we follow the same philosophy as (6) to predict whether a root user is hateful or not. For further tasks, we call this as $dataset_{hate}$.

Our second dataset is the COVID-19 dataset. We use the dataset as described by (3). As our focus is misinformation, we choose the second half of the year 2020, particularly the month of November for our analysis. We randomly sample a day and download around 4 million tweets. We then clean and preprocess the tweets. Further, we create a retweet graph as we have users who have retweeted tweets. This significantly reduces the graph size. Then, we use hashtags which denote misinformation such as those for anti-vaccinations, anti-mask etc. This gives us an equivalent graph structure which we use. We use the Twitter API using the twarc library on Python for downloading the data. Custom regex functions are used to clean and pre-process the data. We particularly remove the user mentions, excessive punctuation and emojis. For further tasks, we call this as $dataset_{covid}$.

We call the FakeNewsNet (13) dataset as $dataset_{base}$. A comprehensive description of our dataset can be found in Table 1.

2.2 Augmenting the news embedding with a document encoder

Here, we create a document encoder using the doc2vec approach as described in the paper (9). We use the source URLs from the FakeNewsNet (13) dataset. We note that because this dataset is a few years old, many news articles are not available at their corresponding URL. To overcome this, we use 2 sources - common crawl and they wayback machine using their corresponding APIs. Using this technique, we are only left with $10 - 15\%$ URLs which do not have their corresponding pages. We use the gensim package to train a model to give a document embedding of the text HTML output of the URLs. We then concatenate this embedding with the original news embeddings. We set the dimension of the document embeddings to 300.

2.3 GATs and Hypergraph Convolution Networks

- **Graph Attention Networks (GAT):** GAT comprises of a novel neural network architectures that operate on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations.It employs self attention over the node features to do so.
- **HyperGraph Convolution Network:** In general, Hypergraph refers to the graph structure in which an edge can join any number of vertices, while in an ordinary graph an edge connects only two vertices. Hypergraph convolution defines a basic convolutional operator in a hypergraph. It enables an efficient information propagation between vertices by fully exploiting the high-order relationship and local clustering structure therein. Graph Convolution is a special case of hypergraph convolution when the non-pairwise relationship degenerates to a pairwise one.

A generic GCN graph VAE model is shown in Figure 2. Here, we pass in the adjacency matrix and the user textual features as the user feature matrix. We use the pytorch geometric library for our experiments.

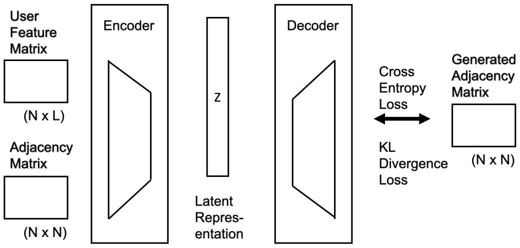

Figure 2: Generic inputs and architecture of a graph auto-encoder

2.4 Historical tweet information inference

We utilize the past information of the users using the latest 200 tweets that we have for $dataset_{hate}$ and $dataset_{base}$. For the $dataset_{covid}$, we mine the latest 200 tweets for the selected users using the twarc library for python. Then, we use the proportion of these tweets which are hate speech/misinformation according to the task. To do this, we use the Perspective API [1] for classifying hate speech for $dataset_{hate}$. We rely on the hashtags for misinformation - as we that is often a good weak indicator of the opinion of the tweet. Once we have the proportion values for each user, we use this to weigh the textual user embedding. After weakly classifying the tweets, we follow the same approach as described in the UPFD (6) paper. We use BERT based embeddings for this task.

Table 2: Hashtags used for hate speech and misinformation

Task	Hashtags
Hate Speech	#trumptrain, #lockherup, #sendthemback
Misinformation	#manmadevirus, #masksdontwork, #maskoff

3 Experiments and Discussions

In this section, we aim to answer the following research questions:

- **RQ1**: Does the base UPFD (6) architecture perform well on the hate speech detection task?
- **RQ2**: What is the effect of adding document embeddings to the misinformation detection task?
- **RQ3**: What is the effect of different graph encoding techniques such as GATs and hypergraph convolutional techniques over conventional GNN methods?
- **RQ4**: How does weighting the user history textual information affect performance?

We summarize our results in Table 3.

Dataset	GCN	GAT	Document Embedding
$Dataset_{base}$ POL	83.1	80.44	82.2
$Dataset_{base}$ GOS	95.2	93.74	94
$Dataset_{hate}$	88.65	85.57	-
$Dataset_{covid}$	75.45	69.35	-

Table 3: F1 scores for our experiments

3.1 RQ1: Does the base UPFD (6) architecture perform well on the hate speech detection task?

Not so surprisingly, we get pretty high scores in determining if a user is hateful or not using the UPFD framework. We train the model using a 70:30 training to test split - given that we have around

[1] https://www.perspectiveapi.com

5,000 labelled and around 543 ground truth labelled hateful users and rest as non hateful. The graph encoder captures the position of the users and the textual information captures whether the past tweets are hateful or not. Thus, we see F1 scores around 0.8 for this task.

3.2 RQ2: What is the effect of adding document embeddings to the misinformation detection task?

We see that adding document embeddings gives only a small increase in the overall performance for the misinformation task. This might be because of the sheer noise in parsing HTML files. Though we have used proper libraries such as beautifulsoup in python, parsing a HTML page requires knowledge of where the content is stored for each URL page. Further, because of the nature of news websites, there is a lot of information not pertaining to the current article but of other linked articles which are on the main page as well. This leads to the document embeddings being more *generic* for a lot of the URLs.

3.3 RQ3: What is the effect of different graph encoding techniques such as GATs and hypergraph convolutional techniques over conventional GNN methods?

Using the UPFD twitter dataset, the model using conventional GCN Layers seems to be taking lowest time and the loss function also seems to be converging faster in comparison to models having 'GAT' and 'HyperConv' layers. Moreover, The loss function of the model with Graph attention layers (GAT) seems to be converging much slower (taking the maximum time in model training) and possesses least accuracy. The models with 'GCN' and 'HyperConv' layers respectively are performing well and have almost same accuracy after certain number of epochs.

3.4 RQ4: How does weighting the user history textual information affect performance?

Weighting the user history leads to significance increase in the performance. This can be explained with the fact that weighing the textual embeddings is a direct measure of the propensity of a user to retweet a similar tweet (misinformation/hate speech). This is in line with results from many papers (12; 6). The hypothesis is that if a user has a proclivity towards producing a certain kind of information/inclined towards an agenda, then there is high chance that they will retweet a tweet containing a similar information, even if it is associated with misinformation/hate speech.

4 Conclusion and Future Work

In this paper, we have presented a generic framework for predicting malicious information using both the textual features ass well as the social graph based features arising from the social interactions of the information. We augment the UPFD architecture by adding 3 important modules which increase the performance of existing models. We further elaborate on the importance of weighting the user textual features. This issue is especially important in today's age as most of the information that we consume and produce comes from social networks. Recent events like the COVID-19 pandemic have further fueled the spread of misinformation. The severity of this issue lead the World Health Organization (WHO) to name this as the Infodemic [2].

This paper gives a direction to future research work that encompasses a holistic view of malicious information detection. As future work, we would like to improve the quality of our labelling techniques. We can do this by devising a crowd sourcing technique. Further, maintaining an active bank of misinformation could be greatly helpful. Further, getting high quality document embeddings is sure to aid this task. This would require us to know about the structure of the external document HTML structure.

The link to our GitHub repository can be found here: `https://github.com/maheshjindal/social_networks_misinformation_detection`

[2] `https://www.who.int/health-topics/infodemic#tab=tab_1`

References

[1] Izzat Alsmadi, Kashif Ahmad, Mahmoud Nazzal, Firoj Alam, Ala Al-Fuqaha, Abdallah Khreishah, and Abdulelah Algosaibi. Adversarial attacks and defenses for social network text processing applications: Techniques, challenges and future research directions. 10 2021.

[2] Song Bai, Feihu Zhang, and Philip H. S. Torr. Hypergraph convolution and hypergraph attention, 2020.

[3] Juan M. Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, Ekaterina Artemova, Elena Tutubalina, and Gerardo Chowell. A large-scale covid-19 twitter chatter dataset for open scientific research—an international collaboration. *Epidemiologia*, 2(3):315–324, 2021. ISSN 2673-3986. doi: 10.3390/epidemiologia2030024. URL `https://www.mdpi.com/2673-3986/2/3/24`.

[4] Cody Buntain and Jennifer Golbeck. Automatically identifying fake news in popular twitter threads. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 208–215. IEEE, 2017.

[5] Shruti Dhapola. Coronavirus vs misinformation: What google, instagram, youtube, facebook and others are doing. Mar 17 2020.

[6] Yingtong Dou, Kai Shu, Congying Xia, Philip S. Yu, and Lichao Sun. User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2051–2055, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462990. URL `https://doi.org/10.1145/3404835.3462990`.

[7] Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. The future of false information detection on social media: New perspectives and trends. *ACM Comput. Surv.*, 53(4), July 2020. ISSN 0360-0300.

[8] Md Rafiqul Islam, Shaowu Liu, Xianzhi Wang, and Guandong Xu. Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, 10, 12 2020. doi: 10.1007/s13278-020-00696-x.

[9] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.

[10] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 3818–3824. AAAI Press, 2016. ISBN 9781577357704.

[11] Sushree Panigrahi and Jeet Singh. Deadly combination of fake news and social media. *Rajiv Gandhi Institute for Contemporary Studies*, 4, 2017.

[12] Manoel Horta Ribeiro, Pedro H. Calais, Yuri A. Santos, Virgílio A. F. Almeida, and Wagner Meira Jr au2. "like sheep among wolves": Characterizing hateful users on twitter, 2018.

[13] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media, 2019.

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[15] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018. doi: 10.1126/science.aap9559.

[16] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, Jan 2021. ISSN 2162-2388. doi: 10.1109/tnnls.2020.2978386. URL `http://dx.doi.org/10.1109/TNNLS.2020.2978386`.

[17] Xinyi Zhou and Reza Zafarani. A survey of fake news. *ACM Computing Surveys*, 53(5):1–40, Oct 2020. ISSN 1557-7341. doi: 10.1145/3395046. URL `http://dx.doi.org/10.1145/3395046`.