Fake News Detection using Graph Representation Learning

Vinita Chappar
vchappar@buffalo.edu
UB person#: 50337016
University at Buffalo

Charanteja Kalva
charante@buffalu.edu
UB Person#: 50336873
University at Buffalo

Abstract

With the rapid expansion of Social media platforms, people also depend on these platforms for news and daily activities happening around the world. As there are no regulations or restrictions on these platforms people have enough freedom to post irrespective of the authenticity of the post. This may result in the wrong influence on the people regarding the news in the post. This gives us the need to find such fake news and eradicate them before it reaches a vast number of audience. So we decided to counter these fake news by some deep learning methods. As we can observe that social media platforms represent graphical networks, we explored some Graph Neural Networks to detect the Fake-News based on the content and the users responsible for its spread. We have used Graph Convolutional Networks and Graph Attention Networks to classify Fake NEWS and got good results, which show that while detecting fake news the Graph network structure play an important role.

1. Introduction

The World is accelerating with progressive developments in technology of all forms. Because of the easy availability of internet and high-speed bandwidth, usage of mobile phones and other related equipment has seen a sudden spike. Subsequently the social media platforms saw the opportunity to keep people engaged to their platform by providing various appealing features. People are now reliable on social platforms for their daily dose of NEWS. Social media platforms provide enough freedom to the users without restricting the type of content the users post. Here comes the problem of FAKE NEWS. The problem with the NEWS articles on social platforms is that the users post the article without any authenticity, which draw in many fake and misleading news. These NEWS traverse through the network because of re-sharing and re-posting by other users who are not even aware of the context of the situation.

In this project we propose a model to identify the fake news using the graph representation of how the news articles are connected and its diffusion process through the social media.

Fake NEWS detection is an emerging topic that has received a lot of attention since last five years. Researchers are continuously trying to make progress in this field of study. Some well-structured datasets are available and different approaches were developed to handle the problem. We adopted the Graph Convolutional Network(GCN) to classify news articles into fake and real classes. News articles are posted by a user and are re-posted and re-shared by other users and the chain continues forming a graph structure between users and news articles. The Fake article takes birth at a node and diffuses through the network. We assume that GCN has the capability to capture this diffusion process helping to improve the performance.

2. Related Works

There are various approaches that can be used to solve the problem of Fake News, some of them are as follows[8]. **Content-based** approaches rely on the text content to detect the truthfulness of news articles, which usually refer to long text. A variety of text characteristics are investigated for supervised learning, including TF-IDF and topic features, language styles, writing styles and consistency and social emotions use recurrent neural networks to learn better representations of user responses. **User-based** approaches model the traits of users who retweet the source story, extract account-based features. Building Heterogeneous graph embedding method is proposed to learn the traits of users so that they can be identified in similar accounts. **Structure-based** approaches leverage the propagation structure in the social network to detect fake news. one method is to leverage the hidden information, i.e., hashtags and URLs, to connect conversations, and find such implicit info can improve the performance of rumor classification. Tri-Relationship method proposed by Kai Shu[4], considers the three auxiliary information available in the social media regarding the news, the source and users re-posting/re-sharing the article. User-User Social Relationship, User Credibility and New User Engagement are the three requirements to capture the relationships. TriFN achieves average relative improvement of 9.23%, 8.48% on BuzzFeed, comparing with LIWC+Castillo [4]. To capture the structure of the network, below mentioned approaches are the good fit.

2.1. Graph Convolutional Network

Because of good learning capacity of CNN's, they were given more priority to be applied to graph structured data. Kief and Welling proposed a GCN model using first-order approximation of localized spectral filters on graphs. Later Hamilton et al. Proposed GraphSAGE to convert this transductive setting into an inductive setting to generalize to the unseen nodes that are not used for training the model. GraphSage algorithm has been considered as base approach to many applications. The algorithm is as follows.

```
Input  : Graph $\mathcal{G}(\mathcal{V},\mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth $K$; weight matrices
         $\mathbf{W}^k, \forall k \in \{1,...,K\}$; non-linearity $\sigma$; differentiable aggregator functions
         $\text{AGGREGATE}_k, \forall k \in \{1,...,K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$
Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$
$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$ ;
for $k = 1...K$ do
    for $v \in \mathcal{V}$ do
        $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
        $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
    end
    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$
```

Figure 1: GraphSage algorithm for constructing convolutions of nodes.[2]

The intuition behind this algorithm is that at each iteration, or search depth, nodes aggregate information from their local neighbors $\mathcal{N}(v)$, and as this process iterates for K times, nodes incrementally gain more and more information from further reaches of the graph. The aggregate function in the algorithm can be replaced by mean, max, or weighted average or any symmetric function.

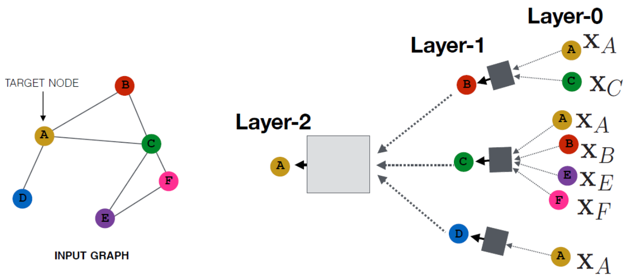


Figure 2: Pictorial representation of constructing node embeddings.[1]

Nodes have embeddings at each layer, current layer embeddings feed on previous layer embeddings and the layer-0 embeddings are the features of individual nodes. Node representation obtained after layer-2 also captures the structure of local graph around the node.

2.2. Graph Attention Network (GAT)

Attention is the mechanism of focusing on the most relevant parts of the input to make decisions. When an attention mechanism is used to compute a representation of a single sequence, it is commonly referred to as self-attention or intra-attention. The key idea behind GAT[11] is to compute the hidden representations of each node in the graph, by attending over its neighbors, following a self-attention strategy. The attention architecture is efficient, since it is parallelizable across node-neighbor pairs, it can be applied to graph nodes having different degrees by specifying arbitrary weights to the neighbors; and the model is compatible to inductive learning problems i.e. can generalise well for unseen data. One of the benefits of attention mechanisms is that they allow for dealing with variable sized inputs.

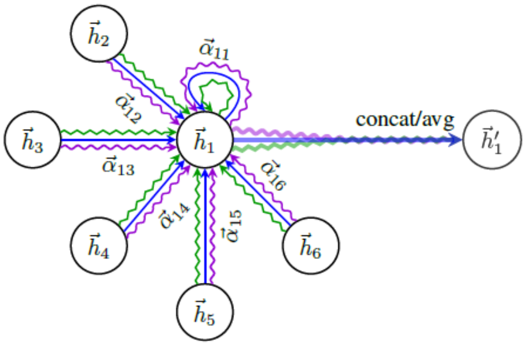


Figure 3: Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain h_1' [11]

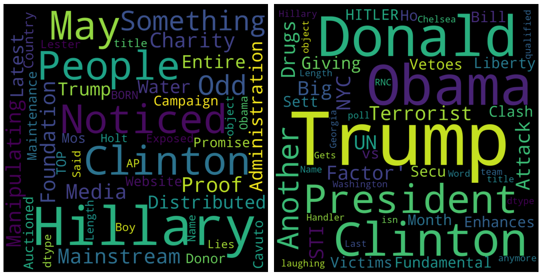
3. Method

3.1. Data Collection

The first step in this project was to collect data. We had to collect fake news and real news data from sources such as facebook, twitter etc. Many data sources are available openly which are a collection of news from such social networks. The selection of the data source was based on these questions:

1. Type of problem to solve: Classification task
2. What data sources already exits: FNID, Twitter15, BuzzFeedNews, Politifact
3. Is the data publicly available: BuzzFeedNews and Politifact are available in abundance

4. Type of the data: Unstructured



(a) Fake News (b) Real news

Figure 4: Highlighting frequent words in the news titles.

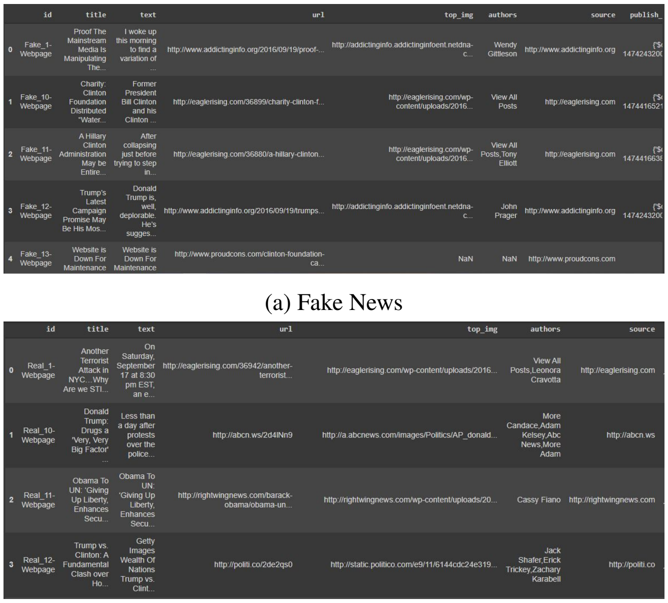
We chose BuzzFeedNews as our datasource since it provides data in a csv as well as json format which makes it easier to process.[9, 10, 7]

If we look at the data and highlight the most evident words using word cloud, the words with higher frequency count/weights will be larger. The fig. 4 shows some of those words for both real as well as fake news. Consider that we don't really know which is the fake news and the real news. If we look at the words, we can't really make out if the news articles are fake, i.e the weights assigned to these words alone is not enough to categorize these articles. That is why we need to extract the text of the news to understand the context behind these titles. This will help us in clearly identifying the news as real or fake. Here is where data pre-processing and NLP comes into play.

3.2. Data Preprocessing

3.2.1 Exploratory Data Analysis

The data was obtained from git repository published by BuzzFeedNews. https://github.com/BuzzFeedNews/2017-12-fake-news-top-50 and the works by [9, 10, 7] The JsON files contain the contents of the news. Along with this, the dataset has a collection of Users and the interaction between users and particular news articles. The content with the JSON key 'text' contains lot of information which could be used to understand the article and associate a label with it: 0 for real news and 1 for fake news.Using this data, we created a data frame using the text and labels.



	id	title	text	url	top_img	authors	source	publish_
0	Fake_1-Webpage	Proof The Mainstream Media Is Manipulating The...	I woke up this morning to find a variation of ...	http://www.addictinginfo.org/2016/09/19/proof-...	http://addictinginfo.addictinginfoent.netdna-c...	Wendy Gittleson	http://www.addictinginfo.org	{'$ 147424320
1	Fake_10-Webpage	Charity: Clinton Foundation Distributed "Water...	Former President Bill Clinton and his Clinton ...	http://eaglerising.com/36899/charity-clinton-f...	http://eaglerising.com/wp-content/uploads/2016...	View All Posts	http://eaglerising.com	{'$ 147441652
2	Fake_11-Webpage	A Hillary Clinton Administration May be Entire...	After collapsing just before trying to step in...	http://eaglerising.com/36880/a-hillary-clinton...	http://eaglerising.com/wp-content/uploads/2016...	View All Posts,Tony Elliott	http://eaglerising.com	{'$ 147441663
3	Fake_12-Webpage	Trump's Latest Campaign Promise May Be His Mos...	Donald Trump is, well, deplorable. He's sugges...	http://www.addictinginfo.org/2016/09/19/trumps...	http://addictinginfo.addictinginfoent.netdna-c...	John Prager	http://www.addictinginfo.org	{'$ 147424320
4	Fake_13-Webpage	Website is Down For Maintenance	Website is Down For Maintenance	http://www.proudcons.com/clinton-foundation-ca...	NaN	NaN	http://www.proudcons.com	

(a) Fake News

	id	title	text	url	top_img	authors	source
0	Real_1-Webpage	Another Terrorist Attack in NYC...Why Are we STI...	On Saturday, September 17 at 8:30 pm EST, an e...	http://eaglerising.com/36942/another-terrorist...	http://eaglerising.com/wp-content/uploads/2016...	View All Posts,Leonora Cravotta	http://eaglerising.com
1	Real_10-Webpage	Donald Trump: Drugs a 'Very, Very Big Factor' ...	Less than a day after protests over the police...	http://abcn.ws/2d4lNn9	http://a.abcnews.com/images/Politics/AP_donald...	More Candace,Adam Kelsey,Abc News,More Adam	http://abcn.ws
2	Real_11-Webpage	Obama To UN: 'Giving Up Liberty, Enhances Secu...	Obama To UN: 'Giving Up Liberty, Enhances Secu...	http://rightwingnews.com/barack-obama/obama-un...	http://rightwingnews.com/wp-content/uploads/20...	Cassy Fiano	http://rightwingnews.com
3	Real_12-Webpage	Trump vs. Clinton: A Fundamental Clash over Ho...	Getty Images Wealth Of Nations Trump vs. Clint...	http://politi.co/2de2qs0	http://static.politico.com/e9/11/6144cdc24e319...	Jack Shafer,Erick Trickey,Zachary Karabell	http://politi.co

(b) Real News

Figure 5: Example Data frame for (a) Fake news dataset and (b) Real news dataset

3.3. Extracting features

Feature extraction is the most critical part of the project. As we know, the text of the news article cannot be fed to a neural network. Thus, we need to represent words or sentences into a vector format which can be fed to our model for training.

3.3.1 Deriving relationships

Once we analyse our data, we can conclude that the relationship between a particular user and the news that they have shared plays a vital role in identifying the main source of the news. With this information, we can categorize the news as fake or real. Here, we are representing this relationship in terms of a bipartite graph. Using this, we can project relationships between two news articles having same users/neighbourhood.The news articles act as the nodes and the edges represent the relation between these articles. This resulting graph can then be represented in the form of an adjacency matrix to feed to the model for training.

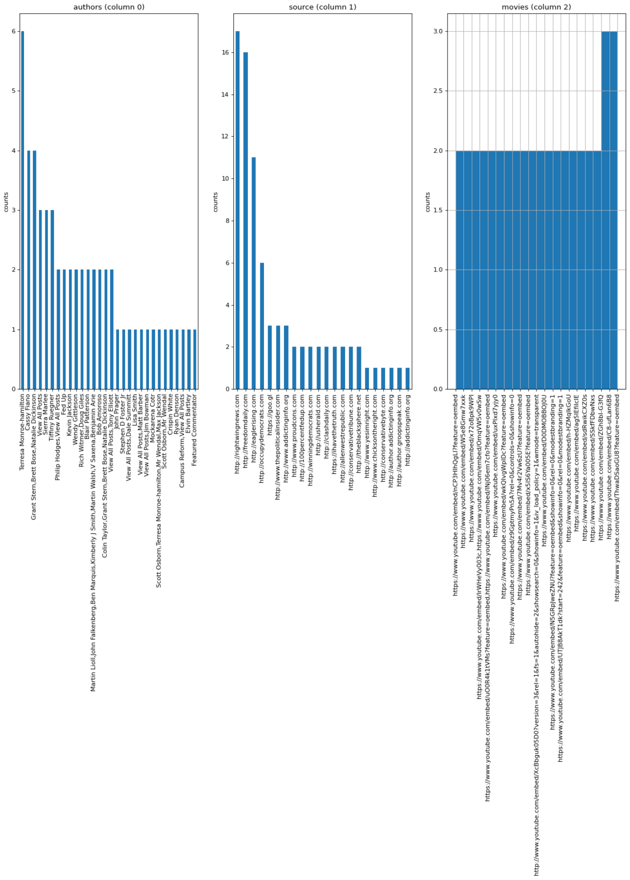


Figure 6: Word frequency count of authors/source/movies for Fake news. By identifying the frequency counts, we can identify from which users/sources most of the fake news comes from.

3.3.2 BERT

The BERT (Bidirectional Encoder Representations from Transformers) is a deeply bidirectional model[3]. It means that BERT learns from both the left and right context in a sentence during the training phase. The BERT model builds on top of transformer blocks and all the transformer layers are "Encoder-only" blocks. [6] Using this functionality of BERT, we tokenize and encode our data into vectors.

BERT provides the users with flexibility of building and training the model from scratch. For this project, we are using the pre-trained model since the data we have is comparatively less than what real world data social networks have.

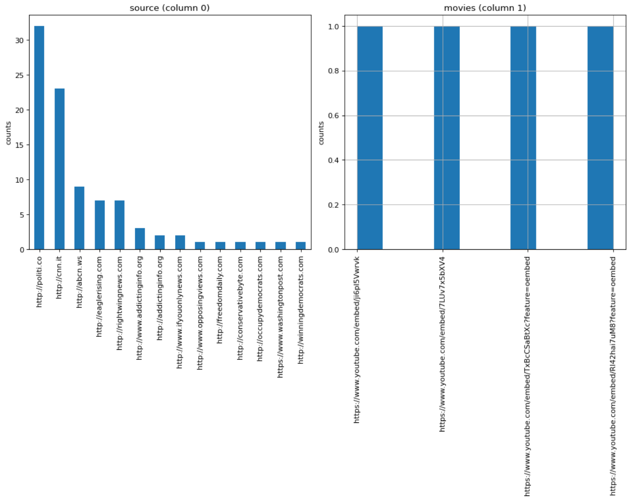
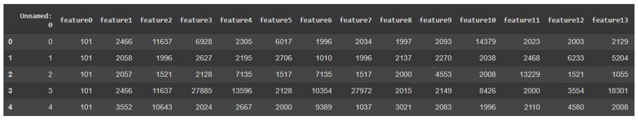


Figure 7: Word frequency count of authors/source/movies for Real news. We can identify which are the frequent sources for real news and enlist them as trustworthy sources for world news.



	Unnamed: 0	feature0	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature13
0	0	101	2466	11637	6928	2305	6017	1996	2034	1997	2093	14379	2023	2003	2129
1	1	101	2058	1996	2627	2195	2706	1010	1996	2137	2270	2038	2468	6233	5204
2	2	101	2057	1521	2128	7135	1517	7135	1517	2000	4553	2008	13229	1521	1055
3	3	101	2466	11637	27885	13596	2128	10354	27972	2015	2149	8426	2000	3554	18301
4	4	101	3552	10643	2024	2667	2000	9389	1037	3021	2083	1996	2110	4580	2008

Figure 8: Sample feature matrix generated after tokenization of the text data into vectors

3.4. Training and Results

We train the models in a semi-supervised manner where we used 60% of the data for training and 20% for testing and 20% as validation dataset.

3.4.1 Graph Convolutional Network

The GCN architecture used is as follows.[5]

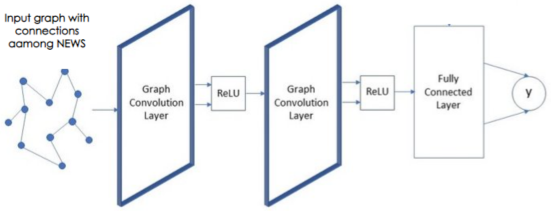
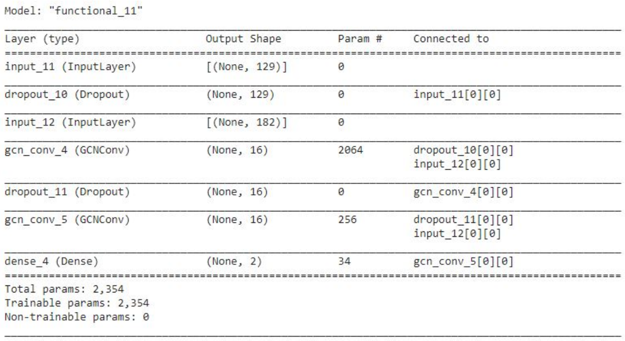


Figure 9: Two layered Graph Convolutional Network.

We observed that by adding an additional dense layer after the second GCN layer improved the performance. We used relu activation and 16 nodes in both GCN layers, used categorical_crossentropy loss function optimized using Adam optimizer. The model summary is as follows.



```
Model: "functional_11"
________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
========================================================================================
input_11 (InputLayer)           [(None, 129)]        0
________________________________________________________________________________________
dropout_10 (Dropout)            (None, 129)          0           input_11[0][0]
________________________________________________________________________________________
input_12 (InputLayer)           [(None, 182)]        0
________________________________________________________________________________________
gcn_conv_4 (GCNConv)            (None, 16)           2064        dropout_10[0][0]
                                                                 input_12[0][0]
________________________________________________________________________________________
dropout_11 (Dropout)            (None, 16)           0           gcn_conv_4[0][0]
________________________________________________________________________________________
gcn_conv_5 (GCNConv)            (None, 16)           256         dropout_11[0][0]
                                                                 input_12[0][0]
________________________________________________________________________________________
dense_4 (Dense)                 (None, 2)            34          gcn_conv_5[0][0]
========================================================================================
Total params: 2,354
Trainable params: 2,354
Non-trainable params: 0
________________________________________________________________________________________
```

Figure 10: Model Summary of GCN architecture used for Fake NEWS classification.

We observed that the model was able to converge for 25 epochs, and was achieving decent results compared to some previous works.

Following is the graph of accuracy measured for 25 epochs. The model was able to achieve an accuracy of **77.7%.**

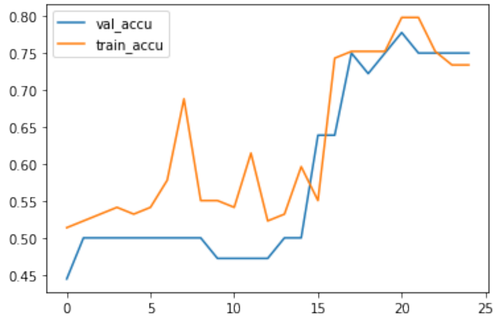


Figure 11: Accuracy for Graph Convolutional Network

Below are the two graphs of training loss and validation loss on the dataset for 25 epochs. We plotted two graphs as the scale of training loss is different from the scale of validation loss. Individually it is easy to observe that both the loss graphs are steadily decreasing with an improvement in the accuracy

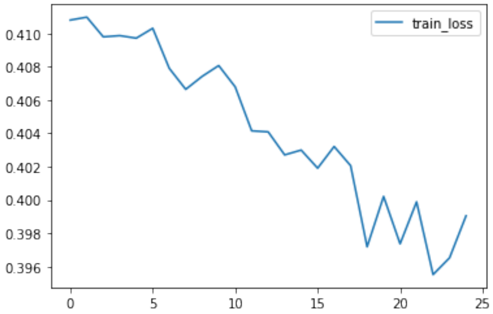


Figure 12: Training loss for Graph Convolutional Network

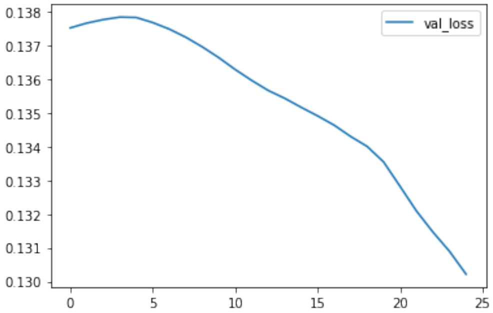
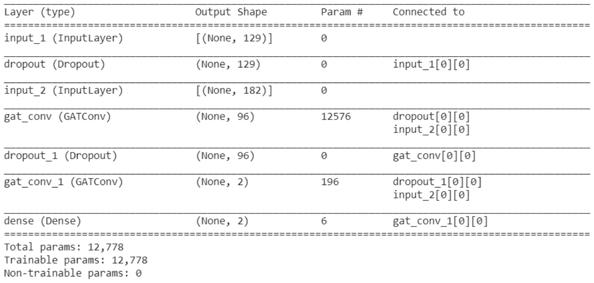


Figure 13: Validation loss for Graph Convolutional Network

3.4.2 Graph Attention Network

Similar to above mentioned architecture we replaced GCN convolution layers with Graph Attention layers. We used two GAT layers followed by a dense connected layer. Below the summary of the model we used for GAT architecture.



```
________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
========================================================================================
input_1 (InputLayer)            [(None, 129)]        0
________________________________________________________________________________________
dropout (Dropout)               (None, 129)          0           input_1[0][0]
________________________________________________________________________________________
input_2 (InputLayer)            [(None, 182)]        0
________________________________________________________________________________________
gat_conv (GATConv)              (None, 96)           12576       dropout[0][0]
                                                                 input_2[0][0]
________________________________________________________________________________________
dropout_1 (Dropout)             (None, 96)           0           gat_conv[0][0]
________________________________________________________________________________________
gat_conv_1 (GATConv)            (None, 2)            196         dropout_1[0][0]
                                                                 input_2[0][0]
________________________________________________________________________________________
dense (Dense)                   (None, 2)            6           gat_conv_1[0][0]
========================================================================================
Total params: 12,778
Trainable params: 12,778
Non-trainable params: 0
```

Figure 14: Model Summary of Graph Attention Network architecture used for Fake NEWS classification.

In the first GAT layer we used 12 channels and 8 attention heads and in the second layer 2 channels with 1 attention head. Each layer has elu activation and used categorical_crossentropy loss function optimized using Adam optimizer.

We can observe that the model was able to learn eventually attaining an accuracy of **83.3%.**

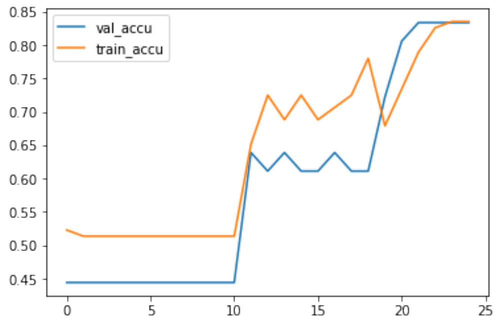


Figure 15: Accuracy for Graph Attention Network

Below are the graphs of Training loss and Validation loss while training the model.

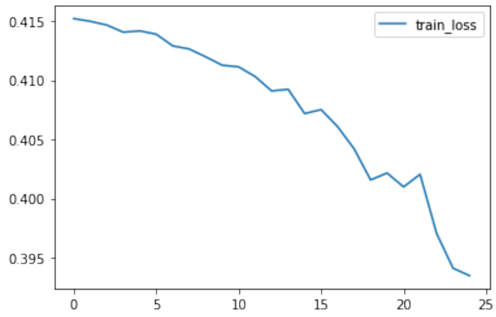


Figure 16: Training loss for Graph Attention Network

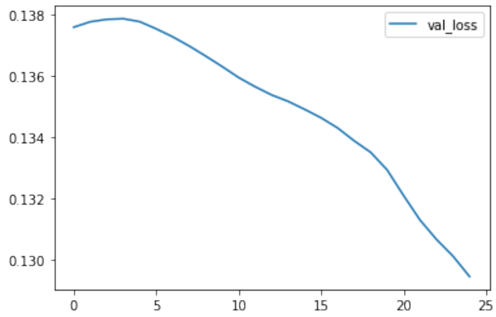


Figure 17: Validation loss for Graph Attention Network

3.5. Analysis

We observed that both the models performed well in terms of capturing the neighborhood structure but GAT outperformed GCN in terms of accuracy. And also the training process in GCN requires more epochs to converge than Attention network. We also tried different combinations of activation functions and found that elu works better with GAT than GCN and continued using relu for GCN.

In GCN it is observed that only 16 filters in convolution layers was sufficient for the dataset we considered and we observed over-fitting of data while we increase the filters. In GAT we used 12 layers and 8 attention heads in first layer making our model avoid over-fitting and generalize well to the test dataset.

4. Future Work

Fake News detection is an important part of NEWS publication, and we observed that there are various aspects that are needed to be considered while classifying the news articles. Apart from the graphical structure of the news, we can consider cross-validating the article once the model has classified it into FAKE. We can use some trusted sources of NEWS and verify the news article with the source. We can even explore the area of determining effective features for detecting fake news from multiple data sources, and also consider news content and social media content. Another notion is to modify the models to consider the users interactions i.e. bipartite graph of users and news. We can develop model that can identify the users that are most responsible for the fake news spread and can flag them as malicious accounts and subsequently deleting such accounts. Build models that can rectify the damage done by fake news that was already spread into the social network by providing the correct information regarding the scenario.

5. Conclusions

Because of the services provided by the social media platforms people are attracted to spend more time on them, and progressively now-a-days people are dependent on these platforms for any type of information starting form the reviews of restaurants to the world wide NEWS. In this context there is a scope for the spread of incorrect information. In this project we have explored two different approaches to achieve the task of Fake NEWS detection. We considered Graphical Neural Network models to elevate the graph structure embedded in the NEWS network and also wanted to capture the diffusion process of news propagation through this network. Through this we can observe that Graph Convolution Network and Graph Attention Networks are capable to solve this by providing reliable results. Moreover, we also observed that GAT out-performs GCN and has potential in this area to evolve into a sturdy system.

6. Individual Contribution:

- Data Collection and Data analysis - Vinita
- Data Pre-Processing - Charan
- Adjacency matrix construction - Vinita

- Graph Convolutional Network Architecture - Charan
- Graph Attention Network Architectures - Vinita
- Experimentation and Analysis - Vinita and Charan
- Hyper Parameter tuning - Vinita and Charan
- Report - Vinita and Charan.

References

[1] Representation learning on networks. *snap.stanford.edu/proj/embeddings-WWW*, 2018. 2

[2] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. 2

[3] Vineet Kumar. Loading bert with tensorflow hub, 2019. 4

[4] Yi-Ju Lu and Cheng-Te Li. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media, 2020. 1, 2

[5] Inneke Mayachita. Training graph convolutional networks on node classification task, 2020. 4

[6] MOHD SANAD ZAKI RIZVI. Demystifying bert: A comprehensive guide to the groundbreaking nlp framework, 2019. 4

[7] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan 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*, 2018. 3

[8] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *CoRR*, abs/1708.01967, 2017. 1

[9] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017. 3

[10] Kai Shu, Suhang Wang, and Huan Liu. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 2017. 3

[11] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. 2