공학석사학위논문

소셜 미디어 속 루머 탐지를 위한 그래프 합성곱 신경망과 어텐션 메커니즘

Dynamic Graph Convolutional Networks with Attention Mechanism for Rumor Detection on Social Media

2020년 8월

서울대학교 대학원
컴퓨터공학부
최지호

소셜 미디어 속 루머 탐지를 위한 그래프 합성곱 신경망과 어텐션 메커니즘

Dynamic Graph Convolutional Networks with Attention Mechanism for Rumor Detection on Social Media

지도교수 김 종 권

이 논문을 공학석사 학위논문으로 제출함

2020년 6월

서울대학교 대학원

컴퓨터공학부

최 지 호

최지호의 공학석사 학위논문을 인준함

2020년 8월

위 원 장 : 전 화 숙 (인)

부위원장 : 김 종 권 (인)

위 원 : 권 태 경 (인)

Abstract

Dynamic Graph Convolutional Networks with Attention Mechanism for Rumor Detection on Social Media

Jiho Choi
Department of Computer Science & Engineering
The Graduate School
Seoul National University

Social media has been a great disseminator for new information and thoughts. Due to its accessibility of sharing information, however, social media has also become an ideal platform for propagations of rumors, fake news, and misinformation. Rumors on social media not only mislead the users of online but also affects the real world immensely. Thus, detecting the rumors and preventing their spread became an essential task. Previous learning-based rumor detection methods adopted to use contents, users, or propagation features of rumors. However, the methods are limited to represent rumor propagation as static graphs, which aren't optimal for capturing the dynamic information of the rumors.

In this study, we propose a novel graph convolutional networks with attention mechanism model named, *Dynamic GCN*, for rumor detection. We first represent rumor posts with their responsive posts as dynamic graphs. The temporal information is used to generate a sequence of graph snapshots. The representation learning on graph snapshots with attention mechanism captures both structural and temporal information of rumor spreads. The conducted experiments on two real-world datasets demonstrate that our model, *Dynamic GCN*, achieves superior results over the state-of-the-art models in the rumor detection task.

Keywords:

rumor detection, graph convolutional networks, dynamic graph, attention mechanism, social media

Student number: 2018-21939

Contents

List of Figures

List of Tables

Chapter I

Introduction

Social media has been a great disseminator for new information and thoughts. Due to its accessibility of sharing information, however, social media has also become an ideal platform for propagations of rumors, fake news, and misinformation [1]. Although the definition of rumor may vary by literature, we use the term rumor to indicate messages in which the veracity labels are unknown at the time of diffusion [2] [3]. Rumors on social media not only mislead the users of online but also affect the real world immensely [4]. Thus detecting the rumors and preventing their spread became an essential task.

Early studies in rumor detection focused on understanding the characteristics of rumors [6] [7] and extracting prominent features of rumor from the text content or the users' profiles [7] – [11]. Also, the temporal or linguistic features were significantly considered in [12] – [14]. [15] – [17] elaborate propagation patterns of rumor and show profound results on rumor detection. The manually extracted content-based, user-based, or propagation-based handcrafted features were used to train classical machine learning classifiers such as a decision tree, random forest, or SVMs. However, the limitation of using manually extracted features is that it fails to capture the higher-order patterns of rumors.

To solve the problem of using handcrafted features and avoid the feature engineering efforts, [18] – [21] had adopted neural networks such as RNNs or CNNs. The proposed rumor detection models were able to capture the high-dimensional representation of rumor from the textural contents, user profiles, and propagation structures. Notably, the models of using propagation structure as features, try to represent the skeptical or conflict opinions from the responsive posts.

Recent advent in Graph Neural Networks (GNNs) and its variant like GCN, GraphSAGE, and GAT [22] – [26] have gained a lot of attention. The GNNs have shown promising results in graph inference tasks such as node classification, graph classification, and link prediction. [27] and [28] have successfully adapted GCN and GAT in the rumor detection domain, respectively. However, both models aren't considering the temporal dynamics of the rumor propagation, which only considers the static graph structure of the final state of rumor propagation.

In this study, motivated by the dynamic nature of rumor propagation, we propose a novel graph convolutional network-based model named *Dynamic GCN* to better understand the evolving pattern of rumor propagation. The model includes two distinct ways of representing rumor propagation with graph snapshots: sequential and temporal snapshots. Figure 1 depicts how the propagation tree can be represented with the sequence of snapshots. The details of the representation will be discussed in section 4.1. The extended GCNs capture the spatial representation of rumor posts within a snapshot. And finally, the series of graph snapshot representations are combined with attention mechanisms. We evaluate the proposed model with two real-world datasets and show our model outperformed other state-of-the-art methods.

We summarize the main contributions as follows:

- We propose two distinct ways of depicting dynamic graph by generating two different graph snapshots: sequential and temporal snapshots.
- We propose a novel GCN based rumor detection model that can capture the evolving pattern of rumor propagation.
- Experimental results on two real-world datasets demonstrate our model accomplishes superior results on rumor detection compare to other state-of-the-art methods.

We organize this paper as follows. In section 2, we briefly review the rumor detection methods and our model's fundamental components; GCNs and attention mechanisms. In section 3, we formulate the rumor detection problem with the propagation structure of rumor. In section 4, we introduce our model in detail of snapshot generation, graph convolution networks, readout layer, attention mechanisms, and prediction. In section 5, the details of experiments and performance evaluation are described. And finally, we conclude this work in section 6.

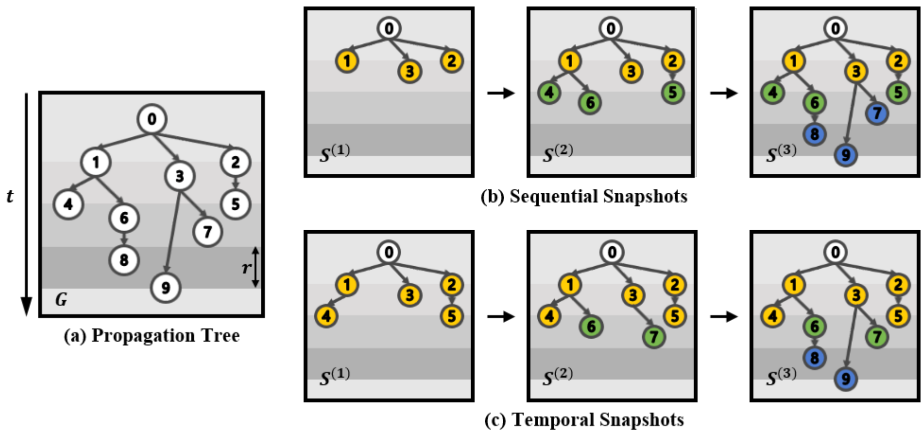

Figure 1. Propagation tree, sequential snapshots, and temporal snapshots

Chapter II

Related Work

2.1 Rumor Detection

Rumor is commonly defined as a message in which the veracity labels are unknown [2] [3]. Rumor detection on social media is a task of classifying messages or posts with their veracity labels. Traditional approaches in rumor detection and other misinformation detection are to extract handcrafted features with prior knowledge on rumors. The content-based method and user-based method were two main approaches [7] – [11]. However, to elaborate on different and additional features, the temporal or linguistic features were considered [12] – [14]. Another characteristic feature of the rumor is its propagation structure. [15] – [17] elaborate propagation patterns of rumor and show profound results on rumor detection. The manually extracted content-based, user-based, temporal, or propagation-based handcrafted features were used to train classical machine learning classifiers such as a decision tree, random forest, or SVMs. However, the limitation of models with handcrafted features is that they fail to capture the high-order patterns of rumors. To solve the problem, [18] – [21] have adopted deep learning models such as RNNs or CNNs variants to extract texture, image features, or user profile features from the rumor posts. Notably, the models utilize propagation structure as features that try to represent the skeptical or conflict opinions from the responsive posts. Recently, sophisticated models like GCN or GAT have successfully adapted in the rumor detection domain.

2.2 Graph Convolutional Networks

Promising results on neural networks in various fields, encourage studies to bring deep learning to topological graph structures. Early studies of node embedding [29], [30] leverage sampling method like random walk for shallow node embedding.

Recent advent in graph neural networks (GNNs) and its variants [22] – [26] made representation learning to be applied directly to a variety of graph structures such as social networks (friendship network, citation network, transaction network), knowledge networks, computer networks, biological graph, chemical graph, and so on. One of the early and honored studies of GNNs is graph convolutional networks (GCNs) [23]. It utilizes the graph Laplacian spectral filters to extend convolutional operations on graphs. Another import study of the GNNs variant is GraphSAGE [24], which proposes different trainable aggregation functions from neighbor nodes embeddings with sampling methods. The proposed aggregation functions like mean, LSTM (random ordered), max-pooling are symmetric, where the ordering of neighbor nodes can be invariant. GAT [25] utilize the attention mechanism for neighbor node embeddings. The GNNs have firmly established the state-of-the-art performance in various graph inference tasks such as node classification, graph classifications, link prediction, and community detection (clustering for the network structure). The fundamental component of GNNs is message passing architecture (or neighborhood aggregation). The key difference in GNN variants is diverse neighborhood aggregation methods and different pooling approaches [31] – [33].

Social networks like graph structure contain the property of dynamics by its nature [34]. Recent studies have proposed different architecture to capture the temporal dynamics of graphs with RNNs and AEs [35] – [37]. Some recent study proposes GCN-based model to capture the dynamic representation of graphs and shows successful results on their applications [38] – [41].

2.3 Learning Sequences & Attention Mechanism

The attention mechanism captures the importance of the input sequence by calculating the attention scores and weights. Compare to LSTM, GRU, like RNNs [42], [43], or Seq2Seq model [44], attention mechanisms have demonstrated outstanding results on both the efficiency and the performance in a variety of fields [45] [46]. Various attention mechanisms have been proposed depending on how they calculate the attention scores and weights. [45] proposed an attention mechanism with additional feedforward neural network to calculate the attention weights. [46] proposed dot-product attention and self-attention, which adapted cosine similarity to capture the importance of certain input words in the task of neural machine translation. Attention mechanism had also introduced in graph representation learning [25].

Chapter III

Problem Definition

Let $C = \{c_1, c_2, \dots, c_m\}$ be the set of m claims, where each claim (or a conversational thread) c_i is consists of n_i microblog posts $P_i = \{p_{i0}, p_{i1}, \dots, p_{i(n_i-1)}\}$. The p_{i0} is the root post of c_i and $n_i - 1$ responsive posts are in sequential order by their post time. The post p_{ij} is represented with F dimensional feature $x_{ij} \in R^F$.

Propagated from the root post, responsive posts form a propagation tree $G_i =< V_i, E_i >$, where each edge represents its direct responsiveness [14] [15]. The vertex set V_i is represented with the posts' features $\{x_{i0}, x_{i1}, \dots, x_{i(n_i-1)}\}$ and the edge set E_i represents set of directed edges from source posts (root or responsive posts) to their direct responsive posts. A_i is an adjacency matrix for the directed graph G_i and $X_i = \left[x_{i0}^T;\ x_{i1}^T;\ \dots;\ x_{i(n_i-1)}^T\right]^T \in R^{n_i * F}$ is a feature matrix for posts P_i. Upon representing the propagation tree as a static graph, to elaborate its evolving pattern, we define the diffusion graph with T step series of graph snapshots $S_i = \left\{S_i^{(1)}, S_i^{(2)}, \dots, S_i^{(T)}\right\}$. The detail of the snapshot formulation will be discussed at section 4.1.

Each claim c_i is associated with its veracity label y_i, where y_i belong to one of four classes $\{T, F, U, N\}$ (True rumor, False rumor, Unverified rumor, or Non-rumor) [15] [20]. In this study, we define the task of rumor detection as a supervised graph classification problem, which the goal is to learn a mapping function $f: C \rightarrow Y$ to classify the veracity labels of c_i using S_i and X_i. The Table 3.1 shows the notations with explanations.

Table 1: Notation

Notation	Explanation
C	set of m claims
c_i	i^{th} claim
P_i	set of posts for the claim c_i
p_{i0}	root post of the claim c_i
$\{p_{i1}, p_{i2}, \dots, p_{(n_i-1)}\}$	set of responsive posts of the claim c_i
x_{ij}	feature vector of post p_{ij}, $x_{ij} \in R^F$
G_i	rumor propagation graph (tree) of the claim c_i,
	$G_i =< V_i, E_i >$
X_i	node feature matrix of the claim c_i, $X_i \in R^{n_i * F}$
T	snapshot count
S_i	set of T snapshots for the claim c_i

Chapter IV

Dynamic GCN with Attention Mechanism

In this section, we propose a dynamic graph representation learning-based model for rumor detection, named *Dynamic GCN* (DynGCN). The main components of the model are snapshot generation, graph convolutional networks, readout layer, and an attention mechanism. Each responsible for rumor propagation representation, representation learning on a graph snapshot, node embedding aggregation for global graph representation, and sequential learning from a series of graph snapshots. Figure 2 is the overview of our dynamic rumor detection model, and Figure 3 is the layers with its input shapes.

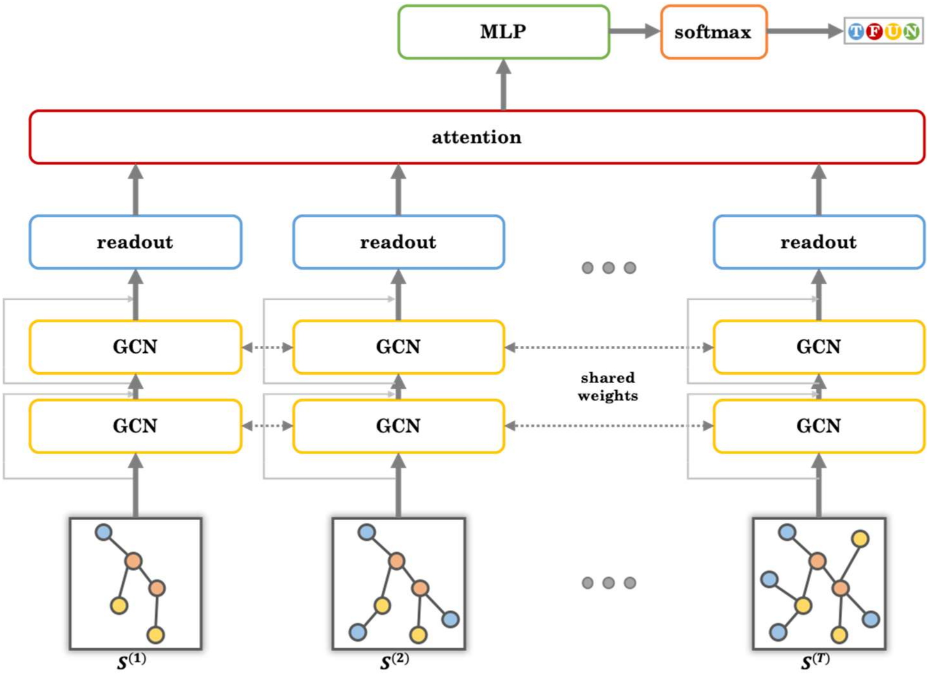

Figure 2. Overview of the *Dynamic GCN* rumor detection model

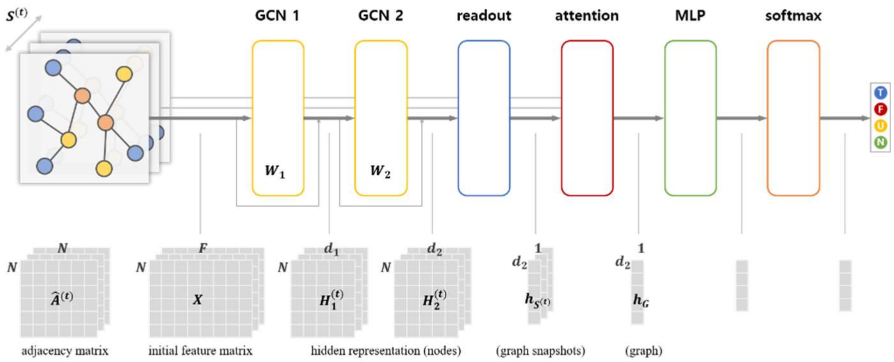

Figure 3. Layers of the *Dynamic GCN* rumor detection model with its dimensions for each layer

4.1 Snapshot Generation

To capture the evolving pattern of the rumor diffusion, we adapt to use the series of graph snapshots. We introduce two different ways of depicting the dynamic graphs as T step graph snapshots $\mathrm{S} = \left\{S^{(1)}, S^{(2)}, \ldots, S^{(T)}\right\}$. One is with sequential snapshots, and the other is with temporal snapshots. In Figure 1, we illustrated the two different methods of snapshot generations. Here on the index, i for the claim c_i will be omitted. $S^{(t)}$ is the graph snapshot at the time step t. Each graph snapshots in S will have separate adjacency matrices $A = \{A^{(1)}, A^{(2)}, \ \ldots, A^{(T)}\}$ with $S^{(t)} =< V^{(t)}, E^{(t)} >$.

Sequential snapshots consider the ordering of the additional nodes and links of the propagation tree. Starting from $S^{(1)}$, the following graph snapshots will contain $\left\lceil \frac{n-1}{T} \right\rceil$ additional links (and nodes), where $n-1$ is the total number of responsive links. Eventually, each graph snapshot $S^{(t)}$ will contain $\left\lceil t * \frac{n-1}{T} \right\rceil$ links. The edge set for the sequential snapshot $S^{(t)}$ is as:

$$E^{(t)} = \left\{e_1, e_2, \ \ldots, e_{\left\lceil t * \frac{n-1}{T} \right\rceil}\right\} \tag{1}$$

Temporal snapshots consider temporal information of the propagation tree. Instead of each snapshot contains the same additional edges, temporal snapshots separate T step diffusion with the fixed time interval r. Time interval r is retrieved by dividing the time difference of the first and the last responsive posts with time step T. The edge set for the temporal snapshot $S^{(t)}$

is as:

$$E^{(t)} = \left\{e \middle| \, \tau_e - \tau_{e_1} \leq r \, * \, (t), \, e \in E\right\},$$
$$r = \frac{\left(\tau_{e_{(n-1)}} - \tau_{e_1}\right)}{T} \qquad (2)$$

, where τ_e is the timestamp of link e and r is the time interval of the snapshots.

4.2 Graph Convolutional Networks

For the snapshot representation learning, we adopted graph convolutional architecture. Upon generating the graph snapshots $S = \left\{S^{(1)}, S^{(2)}, \ldots, S^{(T)}\right\}$ and their adjacency matrices $A = \{A^{(1)}, A^{(2)}, \ \ldots, A^{(T)}\}$, we conducted representation learning on the graph snapshots with the graph convolutional networks (GCNs) [22] – [25].

Introduced in [23], the approximated normalized graph Laplacian [48] is used for high-level node representation learning. Together with adjacency matrix $A^{(t)}$ and feature matrix X, learnable parameters $W_k \in R^{d_{k-1} * d_k}$ are trained, where k^{th} layer produce H_k node embeddings. The GCN model that we adapted is as:

$$H_k = \sigma\left(\hat{A} H_{k-1} W_k\right),$$
$$\hat{A} = \widetilde{D}^{-\frac{1}{2}} \tilde{A} \widetilde{D}^{-\frac{1}{2}}, \qquad \tilde{A} = A + I_N, \qquad \widetilde{D}_{ii} = \sum_j \tilde{A}_{ij} \qquad (3)$$

Trainable parameters W_* are shared between same level of GCNs with different snapshots steps. We used 2-layer GCNs with ReLU as activation function σ. We also adapted skip-connection [49] like method called root feature enhancement [27] to enhance the information from the certain node, in this case the root node. The root representations in previous GCN layer bypass to the next layer as:

$$\widetilde{H}_k = concat(H_k, (H_{k-1})_{root}) \tag{4}$$

And finally, inspired and introduced by [27] [50] [51], instead of perceiving diffusion pattern as undirected graph, we adapted bi-directional GCNs which consider both direction of graph representation separately as:

$$\widetilde{H}_k = concat\left(\overrightarrow{H}_k, \overleftarrow{H}_k\right) \tag{5}$$

4.3 Readout Layer

After the GCN layers embed node representation $H_k^{(t)} \in R^{n*d_k}$ of each graph snapshot $S^{(t)}$, the global graph pooling method is used to convert node embedding to graph embedding. The permutation invariant (symmetric) down-sampling method like max/mean/sum-pool, or even sophisticated pooling method like [31], [33] can be used for the aggregation function in readout layer. In this work, we empirically selected the mean-pooling for global graph pooling. The element-wise mean operation of node embeddings for the global graph snapshot embedding is as:

$$h_{s^{(t)}} = MEAN\left(H_k^{(t)}\right) \quad (6)$$

for the global graph snapshot embedding $h_{s^{(t)}}$.

4.4 Attention Mechanism

To apprehend the dynamic (temporal) information of graph snapshots, we use attention mechanisms. We adapt two well-known attention mechanisms: additive attention [45] and scaled dot-product attention [46]. From the graph snapshot embeddings $h_s = \left\{h_{s^{(1)}}, h_{s^{(2)}}, \dots, h_{s^{(T)}}\right\}$, the goal is to learn the attention weights and use them to combine the weighted inputs.

$$\begin{gathered} Attention(Q, K, V)_{Additive} = Softmax\big(FNN(Q; K)\big)V \\ Softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, z = (z_1, z_2, \dots, z_K) \in R^K \end{gathered} \quad (7)$$

Inspired by [20] [45], for the **additive attention**, we retrieve the context vector m_s of the snapshot sequences by element-wise mean operation of embeddings of h_s. The context vector m_s is used as a query (Q) of the attention mechanism, and h_s is used for the key (K) and value (V). For the additive attention, query and key are concatenated and fed to a feed-forward neural network to calculate attention scores s. The softmax results in attention scores of each snapshot sequences become attention weight α. The weights of additive are calculated as Equation (7).

$$Attention(Q,K,V)_{dot\_product} = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$
$$Softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, z = (z_1, z_2, \dots, z_K) \in R^K \tag{8}$$

Scaled dot-product attention consider cosine similarity of the embeddings when calculating the attention scores and weights. As in Equation (8), we adapt self-attention which the query, key, value is all h_s. The softmax result of similarity measures of snapshots is applied for updating the embeddings.

The outputs of the two different attention layers are both the weighted sequences of the snapshot embeddings. The weighted sum of the snapshot embeddings is used to retrieve h_G, which is the global graph embedding.

4.5 Training & Prediction

For the graph classification task, the graph embedding h_G is fed to the multi-layer perception as:

$$\hat{y} = Softmax\big(MLP(h_G)\big) \tag{9}$$

, where $\hat{y} \in R^{|class|}$ is the probabilities of veracity labels class $= \{T, F, U, N\}$.

$$\mathcal{L} = \sum_{i=1}^{|class|} -y_i^{\mathrm{T}} \log \hat{y}_i \tag{10}$$

Our supervised graph classification model is trained with the cross-entropy loss of the predictions and ground truth veracity labels. Equation (10) is the loss function of our model.

Chapter V

Experiments

In this section, we perform experiments on two real-world datasets and compare the performance of the proposed model, *Dynamic GCN*, with other rumor detection baselines. Further, we conduct ablation studies and analyze the results on different snapshot counts and sequential learning method variants.

5.1 Datasets

We evaluated our model with two publicly available rumor detection datasets: *Twitter15* [13], *Twitter16* [16], [18]. Both of the datasets, the *Twitter15* and *Twitter16*, contain rumor propagation trees, where nodes are posts and links are responsive relations, with one of the four ground truth veracity labels (True rumor, False rumor, Unverified rumor, Non-rumor). The detailed statistics of the datasets are shown in Table (1). We used the bag-of-words (BoW) features by selecting the top 5,000 vocabularies for the corpus by TF-IDF; thus, each post initially contains 5,000 features.

Table 1. Statistics of datasets

	Twitter15	*Twitter16*
# of root posts (# of claim, graph)	1490	818
# of users	276,663	173,487
# of posts	331,612	204,820
# of true rumors	372	205
# of false rumors	370	205
# of unverified rumors	374	203
# of non-rumors	374	205
Avg. time length / event	1,337 hours	848 hours
Min./Avg./Max. of posts / event	55 / 223 / 1,768	81 / 251 / 2,765

5.2 Baselines

We compare our *Dynamic GCN* (DynGCN) model with the following rumor detection baseline models.

- DTC [7]: A decision tree-based classifier with handcrafted features of message, user, topic, and propagation to identify the credibility of microblog posting related to trending topics.
- RFC [11]: A random forest-based ranking method that elaborates the handcrafted feature from inquiry phrases of posts. The main features are linguistic patterns.
- SVM-TS [12]: An SVM model that captures the temporal characteristics of social context features of posts.
- SVM-TK [16]: An SVM model with a propagation-based tree kernel that captures structural information and high-order patterns of rumors.
- GRU [18]: An RNN-based model that learns contextual information of relevant posts over time.
- RvNN [21]: A recursive neural network-based model which captures the patterns of a top-down and bottom-up trees
- Bi-GCN [27]: A graph convolutional network-based model which captures propagation patterns with message passing architecture.
- DynGCN (proposed): A graph convolutional network-based model with an attention mechanism to capture temporal dynamics of graph snapshots.

  We haven't included the PPC [20] and GLAN [28] as our baselines since both of the models include crawled user profiles as additional input features, which could be too biased at the time of this work is conducted. The user profile features (such as whether the user is suspended or whether the user is verified) could be too informative because a few years had passed since the initial collection of the datasets. The results could be distorted and might be too much depended on when the user profiles were crawled at this point. However, we compare our model with the state-of-the-art model [27], which considers the posts relations without additional crawled user profiles.

5.3 Experimental Setup & Implementation Details

We conducted 10 runs of 5-fold cross-validation and reported the average accuracies and F1 scores by each label. For the fair comparison, for the models with early stopping method [52] [53] such as Bi-GCN and ours, we randomly split 4-fold of the training set into 80% training set and 20% validation set, which eventually making 16:4:5 splits for train, validation, and test sets. The validation set was used for early stopping with the patience of 10 epochs.

The model has 128 hidden dimensions for a single GCN layer, including root feature enhancement and 256 dimensions for bi-directional representation. We set 2-layer GCNs and used rectified linear units for the non-linearity. We adapt the dropout rate of 0.5 for GCN layers [54] and DropEdge, graph data augmentation method, rate with 0.2 [55]. We train our model with Adam optimization algorithm [56] with the initial learning rate 5e-4 and a maximum of 200 epochs if not early stopped.

Our model is implemented in PyTorch [57] with PyTorch Geometric [58] for the message passing frameworks. For the baseline models, we conduct experiments with the authors' provided codes with the same hyperparameters that were reported, respectively. For the fair comparison, we directly cited (*) some of the metrics already reported in original papers [16] [21] with equivalent experimental settings due to some handcrafted features that are unavailable at the time of the reproduction.

Table 2. Rumor Detection Performance on Twitter15

(c.f. TR, FR, UN, NR: true / false / unverified rumor, and non-rumor)

Model		Twitter 15				
Model		Accuracy	TR F1	FR F1	UN F1	NR F1
DTC [7] *		0.454	0.317	0.355	0.415	0.733
RFC [11] *		0.565	0.401	0.422	0.543	0.810
SVM-TS [12] *		0.544	0.404	0.472	0.483	0.796
SVM-TK [16] *		0.667	0.772	0.669	0.645	0.619
GRU [18]		0.641	0.688	0.634	0.571	0.684
RvNN [21]		0.723	0.821	0.758	0.654	0.682
BiGCN [27]		0.814	0.793	0.811	0.872	0.768
DynGCN w/ additive attention	(S)	**0.824**	0.8652	0.7928	0.7589	0.7772
	(T)	0.818	0.8596	0.7932	0.7610	0.7788
DynGCN w/ dot-product attention	(S)	0.819	0.8708	0.8164	0.7710	0.8003
	(T)	0.821	0.8586	0.8061	0.7654	0.7819

Table 3. Rumor Detection Performance on Twitter16

(c.f. TR, FR, UN, NR: true / false / unverified rumor, and non-rumor)

Model		Twitter 16				
Model		Accuracy	TR F1	FR F1	UN F1	NR F1
DTC [7] *		0.465	0.419	0.393	0.403	0.643
RFC [11] *		0.585	0.547	0.415	0.563	0.752
SVM-TS [12] *		0.574	0.571	0.420	0.526	0.755
SVM-TK [16] *		0.662	0.783	0.623	0.655	0.643
GRU [18]		0.633	0.577	0.715	0.527	0.617
RvNN [21]		0.737	0.835	0.743	0.708	0.662
BiGCN [27]		0.804	0.718	0.787	0.799	0.895
DynGCN w/ additive attention	(S)	**0.838**	0.8737	0.7659	0.7647	0.7166
	(T)	0.838	0.8526	0.7681	0.7804	0.7277
DynGCN w/ dot-product attention	(S)	0.829	0.8730	0.7560	0.7750	0.7080
	(T)	0.835	0.8620	0.7780	0.7870	0.7010

5.4 Performance Evaluations

Tables 2 and 3 summarize the performances of the rumor detection task of our model, DynGCN, with other baselines for *Twitter15* and *Twitter16* datasets, respectively. The reported performances are accuracies, and F1 scores of our model with both the additive attention and the dot-product attention with the snapshot size of 3. (S) and (T) denote for the results of sequential snapshots and temporal snapshots, respectively. The accuracy results of DynGCN with additive attention shows (S) 0.8184, (T) 0.8177 in *Twitter15*, and (S) 0.8383, (T) 0.8379 in *Twitter16*. The accuracy results of DynGCN with dot-product attention shows (S) 0.8194, (T) 0.8206 in *Twitter15*, and (S) 0.8294, (T) 0.8345 in *Twitter16*. Although the model of additive attention with snapshot size of 3 shows the superior results over others, the two attention methods didn't show significant performance differences. However, both variants of our model outperform other state-of-the-art models, Bi-GCN and RvNN, in both datasets. The results indicate that taking account of the temporal information and evolving pattern of rumor propagation is beneficial in rumor detection.

It is demonstrated that the traditional machine learning-based methods with handcrafted features, (DTC, RFC, SVM-TS, SVM-TK), show lower performances compare to other deep learning-based methods (GRU, RvNN, BiGCN, DynGCN). However, SVM-TS and SVM-TK show superior results within the traditional handcrafted methods since these models are able to utilize temporal features. The result tells that it is constructive to consider temporal information of rumor for rumor detection.

Finally, among the propagation-based baselines, graph-based models, DynGCN and Bi-GCN, outperforms other baselines such as RvNN or GRU since graph convolutional networks can better capture the high-level representation of rumor diffusion.

5.5 Ablation Study

In order to see the performance of our model in different settings, we report two ablation studies. One is with different snapshot counts for sequential and temporal snapshots, and the other is with different learning algorithms for combining snapshot sequences.

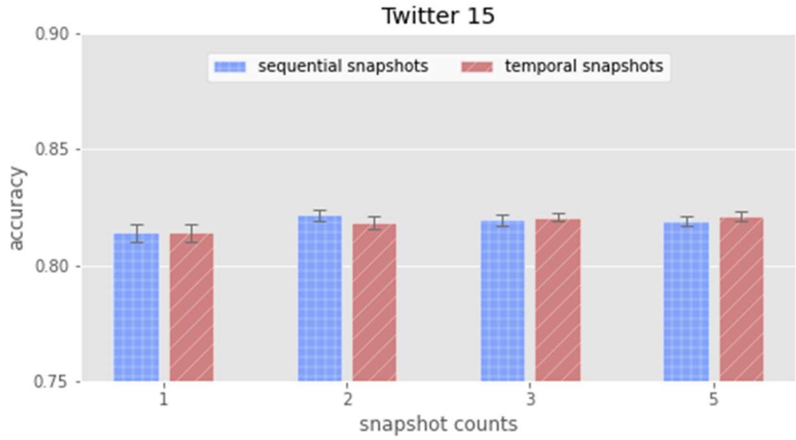

Figure 4. Ablation Study: Snapshot Counts (Twitter15)

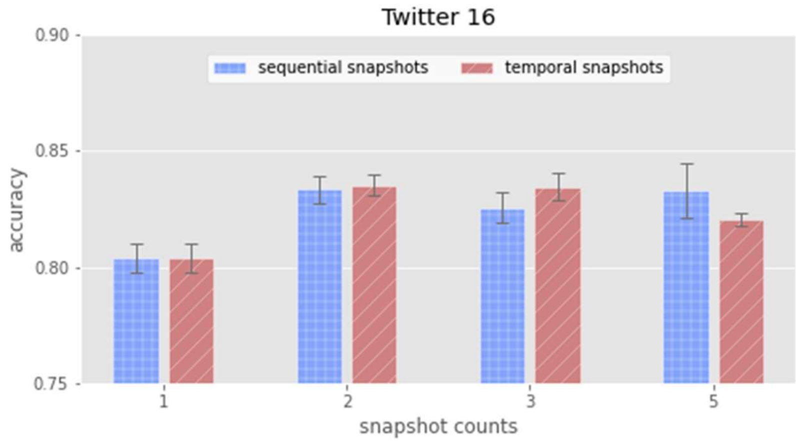

Figure 5. Ablation Study: Snapshot Counts (Twitter16)

Figures 4 and 5 are the results of DynGCN with the snapshot counts of [1, 2, 3, 5] with dot-product attention. Although there aren't significant correlations in the aspects of accuracy with the counts, adapting multiple snapshots show better performance compare to a single static snapshot in both sequential and temporal snapshots.

In both datasets, however, the models with snapshot counts of 2 and 3 show the superior results over other snapshot sizes. We observed that simply applying larger snapshot counts won't produce improved results.

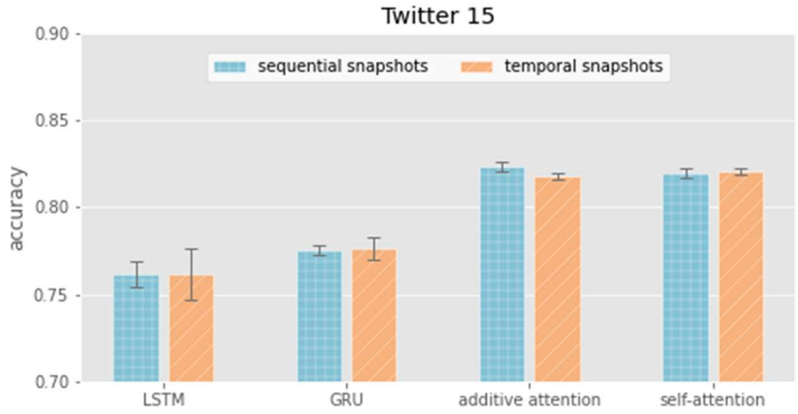

Figure 6. Ablation Study: Learning Sequences (Twitter15)

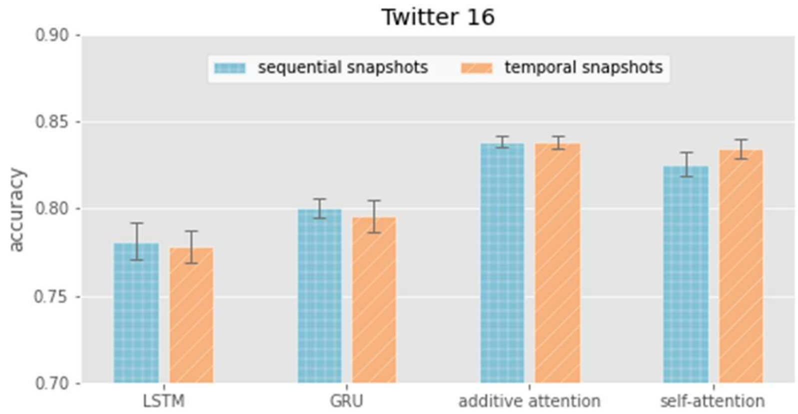

Figure 7. Ablation Study: Learning Sequences (Twitter16)

The attention layer of our model can be replaced with other Seq2Seq [44] models since the inputs to the attention layer are a sequence of snapshot representations. Figures 6 and 7 are the result of different sequence learning methods (Bidirectional LSTM, Bidirectional GRU, additive attention, dot-product attention (self-attention)) with snapshots count of 3. Attention mechanisms which used the weighted sum of sequential or temporal snapshot representations outperform the other RNN-based models. Bidirectional LSTM/GRU show low performance in capturing the temporal representation of graph snapshots. We suspect the results of LSTM and GRU are due to the short sequence of global graph snapshots.

Chapter VI

Conclusion

In this research, we propose *Dynamic GCN*, an end-to-end GCN based rumor detection model with attention mechanisms. The model is able to capture the dynamic of rumor propagations using sequential snapshots and temporal snapshots. The model adapts to use graph convolutional networks to retrieve the embeddings of sequential snapshots and temporal snapshots. The sequences of embeddings are aggregated with the additive attention and dot-product attentions. We empirically evaluate our model with two real-world datasets and compare the performances of rumor detection (veracity classification) with other baseline models. The results show the proposed model outperforms other baseline methods in the rumor detection task. The ablation study reports the performance difference with snapshots counts and the learning sequence variants. We believe there is still room for improvement in the context of GCNs variants, alternative the global pooling method of the snapshots, and additional features from a different context.

Bibliography

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.

[2] Q. Li, Q. Zhang, L. Si, and Y. Liu, "Rumor detection on social media: Datasets, methods and opportunities," in Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 66–75.

[3] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2011, pp. 1589–1599.

[4] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," ACM Computing Surveys (CSUR), vol. 51, no. 2, pp. 1–36, 2018.

[5] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild et al., "The science of fake news," Science, vol. 359, no. 6380, pp.1094–1096, 2018.

[6] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018.

[7] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in Proceedings of the 20th international conference on Worldwide web, 2011, pp. 675–684.

[8] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, 2012, pp. 1–7.

[9] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in 2013 IEEE 13thInternational Conference on Data Mining. IEEE, 2013, pp. 1103–1108.

[10] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 729–736.

[11] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in Proceedings of the 24thinternational conference on world wide web, 2015, pp. 1395–1405.

[12] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1751–1754.

[13] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah, "Real-time rumor debunking on twitter," in Proceedings of the 24th ACM International

Conference on Information and Knowledge Management, 2015, pp.1867–1870.
[14] S. Kwon, M. Cha, and K. Jung, "Rumor detection over varying time windows," PloS one, vol. 12, no. 1, 2017.
[15] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on Sina Weibo by propagation structures," in 2015 IEEE 31st international conference on data engineering. IEEE, 2015, pp. 651–662.
[16] G. W. Ma, Jing and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 708–717.
[17] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in Proceedings of the eleventh ACM international conference on Web Search and Data Mining, 2018, pp. 637–645.
[18] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, ser. IJCAI'16. AAAI Press, 2016, p. 3818–3824.
[19] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, ser. IJCAI'17. AAAI Press,2017, p. 3901–3907.

[20] Y. Liu and Y.-F. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in Thirty-Second AAAI Conference on Artificial Intelligence,2018.

[21] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in Proceedings of the 56thAnnual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1980–1989.

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in Advances in neural information processing systems, 2016, pp. 3844–3852.

[23] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in Proceedings of the 5th International Conference on Learning Representations, ser. ICLR ’17, 2017.

[24] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in neural information processing systems, 2017, pp. 1024–1034.

[25] P. Veliˇckovi´c, G. Cucurull, A. Casanova, A. Romero, P. Li`o, and Y. Bengio, "Graph Attention Networks," International Conference on Learning Representations, 2018.

[26] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in International Conference on Learning Representations,

2019.

[27] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 549–556.

[28] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly embedding the local and global relations of heterogeneous graph for rumor detection," in 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 796–805.

[29] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining,2014, pp. 701–710.

[30] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.

[31] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec,"Hierarchical graph representation learning with differentiable pooling," in Advances in neural information processing systems, 2018, pp. 4800–4810.

[32] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[33] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in 36th

International Conference on Machine Learning, ICML 2019, ser. 36th International Conference on Machine Learning, ICML 2019. International Machine Learning Society (IMLS), Jan. 2019, pp. 6661–6670,36th International Conference on Machine Learning, ICML 2019.

[34] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," ACM transactions on Knowledge Discovery from Data (TKDD), vol. 1, no. 1, pp. 2–es, 2007.

[35] L. Zhou, Y. Yang, X. Ren, F. Wu, and Y. Zhuang, "Dynamic network embedding by modeling triadic closure process," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[36] P. Goyal, N. Kamra, X. He, and Y. Liu, "Dyngem: Deep embedding method for dynamic graphs," arXiv preprint arXiv:1805.11273, 2018.

[37] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "GraphRNN: Generating realistic graphs with deep auto-regressive models," in Proceedings of the 35th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 5708–5717.

[38] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," IEEE Transactions on Intelligent Transportation Systems, 2019.

[39] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in Proceedings of the Thirty-

Fourth AAAI Conference on Artificial Intelligence, 2020.

[40] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dysat: Deep neural representation learning on dynamic graphs via self-attention networks," in Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 519–527.

[41] F. Manessi, A. Rozza, and M. Manzo, "Dynamic graph convolutional networks," Pattern Recognition, vol. 97, p. 107000, 2020.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in NIPS 2014 Workshop on Deep Learning, December 2014, 2014.

[44] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104–3112.

[45] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in3rd International Conference on Learning Representations, ICLR 2015; Conference date: 07-05-2015Through 09-05-2015, Jan. 2015.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998–6008.

[47] A. Zubiaga, M. Liakata, R. Procter, G. W. S. Hoi, and P. Tolmie,"Analysing how people orient to and spread rumours in social media by looking at conversational threads," PloS one, vol. 11, no. 3, 2016.

[48] F. R. Chung and F. C. Graham, Spectral graph theory. American Mathematical Soc., 1997, no. 92.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[50] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural net-works, "IEEE transactions on Signal Processing, vol. 45, no. 11, pp.2673–2681, 1997.

[51] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks, "IEEE Transactions on Neural Networks and Learning Systems, 2020.

[52] L. Prechelt, "Early stopping-but when, "Advances in neural information processing systems, vol. 2, pp. 55–69, 1996.

[53] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning, "Constructive Approximation, vol. 26, no. 2, pp. 289–315, 2007.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from over-fitting," The journal of machine learning research, vol. 15, no. 1, pp.1929–1958, 2014.

[55] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in International Conference on Learning Representations. https://openreview. net/forum,2020.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.

[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antigaet al., "Pytorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.

[58] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

초 록

소셜 미디어 속 루머 탐지를 위한 그래프 합성곱 신경망과 어텐션 메커니즘

최지호
컴퓨터 공학부
서울대학교 대학원

소셜 미디어는 강력한 정보 전달력을 가진 매체로 새로운 정보와 생각의 전파 창구이다. 소셜 미디어의 특징인 접근성은 때론 루머, 가짜 뉴스, 잘못된 정보의 전파에서도 이상적인 플랫폼이 된다. 소셜 미디어 속 루머는 온라인 사용자를 오도할 뿐만 아니라 때론 현실 세계에도 큰 영향을 미친다. 따라서, 루머를 탐지하고 그 전파를 막는 노력이 요구된다. 기존의 루머 탐지 방법은 루머의 내용, 사용자, 또는 전파 과정의 정보를 특성으로 이용한다. 이러한 방법은 루머의 전파를 정적 그래프로 표현하며 그 구조적 특성을 이용한다. 하지만 이는 루머의 동적 특성을 포착하지 못한다는 한계를 가지고 있다. 본 연구에서는 그래프 합성곱 신경망 (graph convolutional networks: GCN)과 어텐션 메커니즘 (attention mechanism)을 활용한 동적 그래프 합성곱 신경망 (Dynamic GCN) 루머 탐지 모델을 제시한다. 먼저, 소셜 미디어 속 루머 게시글들 (posts) 과 그의 답장이 되는 글들(responsive posts)을 이용하여 루머의 전파 과정을 정적 그래프로 표현한다. 시간 정보를 통해 전파 과정을 포함하고 있는 정적 그래프의 집합인 그래프 스냅숏 (graph snapshot) 시퀀스 (sequence)를 만들게 된다. 어텐션 메커니즘을 활용한 그래프 스냅숏 표현 학습은 루머 전파의 구조적 시간적 정보를 모두 효과적으로 반영한다. 실제 트위터 데이터를 이용한 실험을 통하여 제시된 모델의 성능이 다른 비교 모델들보다 높음을 확인할 수 있었다.

주요어 : 루머 탐지, 그래프 합성곱 신경망, 동적 그래프 표현 학습, 어텐션 메커니즘, 소셜 미디어

학 번 : 2018-21939

감사의 글

너무나도 빠르게 지나간 대학원 생활이었지만 지난 2 년 동안 너무 많은 분께 도움을 받아, 이번 졸업 논문을 통하여 감사의 말씀을 전하고 싶습니다.

감사한 분이 너무나도 많습니다. 존경하며 항상 지혜와 배려를 주시는 김종권 교수님. 교수님께서 주신 지도와 지혜, 그리고 배려를 통하여 저는 더 많이 성장할 수 있었던 것 같습니다. 주신 가르침 잘 간직하겠습니다. 감사합니다. 또한 바쁘신 와중에 학위 심사를 위해 귀중한 시간을 내어주신 전화숙 교수님, 권태경 교수님께도 감사드립니다.

연구실 생활 동안 많은 시간을 함께하고 너무나 큰 힘이 되어준 연구실 선후배분들께도 감사함을 전합니다. 재훈이 형, 보연 누나, 서향 누나, 시현 누나, 준현이 형, 형호 형, 지현, 람, 영준이 형, 온유, 건호 형, 태욱, 윤혁, 아현, 형호 소중한 인연들 모두 감사드립니다. 특히 소셜팀에서 함께 연구한 시현 누나, 형호 형, 태욱이, 윤혁이와 동기로써 많은 일을 함께하고 큰 힘이 되어준 온유와 건호 형, 항상 저의 고민에 도움을 주신 재훈이 형, 서향 누나, 시현 누나, 준현이 형, 영준이 형께 감사드립니다. 연구실 생활에 활력을 넣어준 태욱, 윤혁, 아현, 형호 모두 감사드립니다.

대학 친구들, 선후배들, 고등학교, 중학교, 초등학교 친구, 선후배들, 인턴을 하며 만난 인연들을 포함해서 이곳에 미처 적지 못한 감사한 분들이 너무 많습니다. 모두 감사드립니다.

마지막으로 항상 저에게 아낌없는 응원과 사랑을 주시는 사랑하는 우리 가족들에게도 감사합니다. 할머니, 엄마, 아빠, 누나, 매형 모두 많이 사랑하고 감사드립니다.

한분 한분 덕분에 많이 배웠고 행복하게 석사 과정을 마무리할 수 있었습니다. 진심으로 감사드립니다.

2020 년 7 월

최지호 올림