Fake News Detection using Higher-order User to User Mutual-attention Progression in Propagation Paths

Rahul Mishra
University of Stavanger, Norway
rahul.mishra@uis.no

Abstract

*Social media has become a very prominent source of news consumption. It brings forth multifaceted, multimodal and real-time information on a silver platter for the users. Fake news or rumor mongering on social media is one of the most challenging issues pertaining to present web. Previously, researchers have tried to classify news propagation paths on social media (e.g. Twitter) to detect fake news. However, they do not utilize latent relationships among users efficiently to model the influence of the users with high prestige on the other users, which is a very significant factor in information propagation. In this paper, we propose a novel **Hi**gher-order User to User **M**utual-**a**ttention **P**rogression (**HiMaP**) method to capture the cues related to authority or influence of the users by modelling direct and indirect (multi-hop) influence relationships among each pair of users, present in the propagation sequence. The proposed higher order attention trick is a novel contribution which can also be very effective in case of transformer architectures[30]. Our model not only outperforms the state-of-the-art methods on two publicly available Twitter datasets but also explains the propagation patterns pertaining to fake news by visualizing higher order mutual-attentions.*

1. Introduction

Social Media platforms have become part and parcel of our daily lives and are also being used as a common ground for discussions and debates. Rumors and fake news on social media platforms have become a common phenomenon, curbing them is a very challenging and daunting task. The spread of a viral news item or a tweet can seriously affect the election outcomes, reputation of some companies or even relationships among countries, therefore prevailing the sanity in such platforms is the need of the hour. Several machine learning based solutions are investigated in the literature to detect and mitigate the effects of fake news.

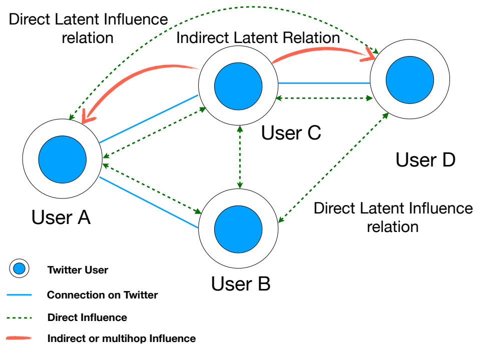


Figure 1. Latent Influence Relationships among users

Previous studies in the literature have used many different facets and aspects related to news items for fake news detection such as the content of the news, source of the news, user response on news and propagation patterns on social media platforms. The news content oriented solutions use handcrafted text or linguistic features to learn a classifier [21, 4, 12], some other works use deep learning techniques instead, to automatically learn the representative features [14, 25]. Recently, neural attention based techniques are also proposed by researchers to detect the misinformation and they also extract evidences pertaining to classifier's decision as a bi-product [22, 18]. Some of the other interesting works use only temporal propagation patterns of the news items on social media to detect the rumors [13, 17]. The advantage of the propagation patterns based methods over news content and user response oriented methods is that they do not rely on user comments and replies as at an early stage of news propagation, these features are not available readily. However, there are some limitations of these approaches, firstly, they use temporal user characteristics such as number of user followers and followings, the numbers of tweets and retweets posted, which requires tedious feature engineering and transformations. Secondly, they do not model the influence or affinity relationships among the users, which is a key factor in information propagation on social media platforms.

We propose a novel Higher Order User to User Mutual-attention Progression (HiMaP) method to address the limitations of existing methods. Rather than using hand-crafted user characteristics features, we use user embeddings, learned via several node embedding techniques. We use user-to-user mutual-attention method to model latent influence relationship among users in propagation paths, which inherently captures the patterns and connotations pertaining to rumor and non-rumor propagation. In figure 1, there are four twitter users A, B, C and D represented as circles and the blue connection lines represent the way they are connected on twitter. Let's assume all of the above mentioned users are the part of a propagation path of a news item n. We compute two kinds of latent influence relationship among the users A, B, C and D. Firstly, we compute direct user to user influence relationship using mutual-attention such as $A \leftrightarrow B$, $A \leftrightarrow C$, $A \leftrightarrow D$, $B \leftrightarrow C$, $B \leftrightarrow D$ and so on, which are depicted as green dotted connection lines in figure 1. Secondly, we compute indirect user to user influence relationship using higher order mutual-attention progression method such as $A \leftarrow C \rightarrow D$, which is depicted as red stroke lines in figure 1.

We use two publicly available twitter datasets for evaluation and analysis, the proposed model outperforms all the baselines and state of the art methods.

In nutshell, major contributions of this paper are:

1. We are the first to use the User to User mutual-attention in propagation paths to model and capture the latent cues related to authority or influence of the users.

2. We enhance the User to User mutual-attention by introducing a novel High Order Mutual-attention Progression method (HiMaP) to model multi-hop latent relationships among the users.

3. Contrary to previous works, we use both the follower and the retweet networks to learn user embeddings rather than representing users with user characteristics vector.

4. We achieve significant gains over state-of-the-art models in terms of accuracy, on two publicly available twitter datasets.

5. We visualize and analyse the attention weights to check the efficacy of the attention mechanism.

2. Related Work

We can categorize the previous works related to fake news detection into three major categories based on what features they utilize, 1. news content and linguistic feature oriented, 2. user action on news oriented and 3. social context oriented. The first category of works use text content of the news items, extract several linguistic and statistical features and learn a classifier to detect whether or not it is a rumor [15, 4, 12, 23, 9]. The authors of [21] use language stylistic feature and source credibility features to model the credibility of web claims. The second category of works use user actions on news, such as sentiments, comments, replies and disapproval. In [34], authors use Bayesian network model (probabilistic graphical model with Gibbs sampling) to capture the conditional dependencies among the truthfulness of news, the users' opinions, and the users' credibility. Authors of [24] propose a CNN based model with a user response generator, which learns to generate a synthetic user response to a news article text from historical user responses, which is used as a user action feature in fake news detection.

The third kind of works utilize social context in terms of user profile, social network features and news propagation paths [15, 27, 26]. The authors of [13] transform the news propagation into a multivariate time series of user characteristics and learn a classifier with concatenated representation of RNN-Based and CNN-Based propagation path representations. Authors in [17] use tree structured neural networks to represent propagation paths, recursive nature of their model effectively captures the tree features of the propagation trees. Authors of [31] propose a new kind of community preserving user embedding method and convert the news propagation tree structures into a temporal sequence and then apply RNN with early stopping for classification.

In contrast to these existing works, HiMaP uses a novel mutual-attention progression model to learn better propagation path representation along with RNN based sequence encoder, which contains cues related to both compositional aspects and latent influence aspects of the propagation sequence.

3. Problem Definition and Proposed Model

3.1. Problem Definition

Given a news item $n \in N$, along with its propagation path on twitter as $u_1 \rightarrow u_2 \rightarrow u_3 .... u_{m-1} \rightarrow u_m$, where u_1 is the user, who has posted the original tweet about news n. u_m is the last user in the sequence, who has retweeted the same tweet. The goal is to classify the news as one of these classes:"True (T)" or "False (F)" or "Unverified (U)" or "Debunking (D)".

3.2. Retweet Propagation Path Representation

We represent the propagation paths of the news by sequence of users pertaining to the original (source) tweets and re-tweets as variable length multivariate time series, very similar to [13]. From the original propagation trees, we create a flattened representation of the tree as a multivariate time series comprising user embeddings and timestamp. For a news item n_i, propagation sequence $Prop(n_i)$

can be defined as:

$$Prop(n_i) =< (f(u_1), t_0)......(f(u_m), t_m) > \tag{1}$$

Where $(f(u_i), t_i)$ represents i^{th} user, i^{th} timestamp and m is the length of propagation sequence. In contrast to [13], we represent each user with learned user embeddings by applying suitable node embedding methods to follower network and retweet network rather than representing users as their characteristics vectors, as depicted in figure 2. Usage of node embeddings instead of characteristics vectors, not only saves the time required to crawl the characteristic features for each user but also does not require any feature engineering.

3.3. Learned User Embeddings

We use unsupervised network representation learning methods to learn user (node) embeddings from both the follower network and the retweet network and we combine the corresponding embeddings for each user by concatenating them. Given a follower graph $F = (V, E)$ and a retweet graph $R = (V', E')$, we compute user embedding $f(v)$ for each user v by concatenating user embedding learned from follower network $f_F(v) \in \mathbb{R}^d$ and the user embedding learned from retweet network $f_R(v) \in \mathbb{R}^d$ as:

$$f(v) = f_F(v) \parallel f_R(v) \tag{2}$$

Provided $v \in V$ and $v \in V'$. Specifically we experiment with DeepWalk [20], Node2vec [8], Line [28] and APP[36] node embedding methods and select the best performing embedding technique.

- **DeepWalk**: It is a uniform random walk simulation based method, which uses SkipGram with hierarchical softmax as optimizer and objective function as follows:

$$min_\phi - \log P(\{v_{i-w}, ..., v_{i-1}, v_{i+1}, ..., v_{i+w}\}|\phi_i) \tag{3}$$

- **Node2vec**: It is a breadth first (BFS) and depth first search (DFS) based method, which uses SkipGram with negative sampling and objective function as follows:

$$max_f \sum_{u \in V} [-\log Z_u + \sum_{n_i \in N_s(u)} f(n_i).f(u)] \tag{4}$$

- **APP**: It is a Personalized PageRank Context based method, which uses negative sampling and objective function as follows:

$$\log \sigma(\vec{s_u}.\vec{t_v}) + k.E_{tn}P_D[\log \sigma(\vec{s_u}.\vec{t_n})] \tag{5}$$

- **Line**: It is a Adjacency matrix based method, which models neighbourhood proximity using negative sampling and objective function as follows:

$$O1 = \sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j) \tag{6}$$

For more details about these node embedding methods, please refer to respective papers.

3.4. LSTM based Propagation Path Sequence Encoder

We use Long short term memory unit (LSTM) [10] to encode the propagation path sequence $f(u_1) \rightarrow f(u_2) \rightarrow f(u_3)....f(u_{m-1}) \rightarrow f(u_m)$ represented as a sequence of learned user embeddings. At a particular time-step t, the current hidden state h_t is computed using standard LSTM equations as follows:

$$\begin{aligned} f_t &= \sigma(W_f.[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i.[h_{t-1}, x_t] + b_i) \\ \tilde{C_t} &= \tanh(W_C.[h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C_t} \\ o_t &= \sigma(W_o.[h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t), \end{aligned} \tag{7}$$

Where, h_{t-1} is previous hidden state and x_t is the current input from input propagation path sequence. We use last hidden state as a compositional representation of propagation path sequence R_C, where

$$R_C = h_m \tag{8}$$

3.5. User to User Mutual-attention

We explain the user to user mutual-attention in detail in this section, as pictorially represented in the figure 2. Neural attention [3] mechanisms are proven to be very effective in many NLP [30, 35, 6] and computer vision applications [32, 7, 11]. The key idea behind the neural attention is to select the important words or sentences in NLP applications and to gauge the crucial areas or blocks in images in typical computer vision applications. Many of the previous works use attention mechanisms to detect fake news[22, 18] and to do fact checking[19]. It is well studied fact that influential users play a very crucial role in information diffusion on social media platforms[2, 1] on the other hand it's very hard to quantify the influence and it's penetration in a real world social network. Interpersonal relationships among users are the key factor in determining the influence [5]. We are the first to propose a User to User mutual-attention method to model the influence among the users. Previously, researchers have used mutual-attention mechanism in case of word to word mutual-attention within a sentence to model intra-relationships among words, present in same sentence [29]. Given a propagation path of a news item n, in terms of sequence of learned user embeddings as $f(u_1) \rightarrow f(u_2) \rightarrow f(u_3)....f(u_{m-1}) \rightarrow f(u_m)$, in the first step we model the relationship among each pair of users present in the propagation path. In very similar fashion to[29], We use a dense layer to project the concatenation of each user embedding pair into a scalar score:

$$S_{ij} = W_{cat}([f(u_i); f(u_j)]) + b_{cat} \tag{9}$$

Where $W_{cat} \in \mathbb{R}^{2d\times 1}$ is a weight matrix, $b_{cat} \in \mathbb{R}$ is bias term and S_{ij} is the latent affinity between users u_i and u_j. Score matrix is $\underset{m\times m}{\mathrm{S}}$ is a square matrix. To compute mutual-attention scores, we can consider two options, either we can apply row-wise max-pooling or row-wise avg-pooling.

$$\begin{aligned} A_C &= Softmax(\underset{row}{max} S) \\ &\qquad or \\ A_C &= Softmax(\underset{row}{avg} S) \end{aligned} \tag{10}$$

Where $A_C \in \mathbb{R}^m$ is the learned attention weight vector. Finally, user to user mutuallly-attended representation R_A of the propagation path can be computed as:

$$R_A = \sum_{i=1}^{m} f(u_i) A_{Ci} \tag{11}$$

3.6. Multi-hop Latent Relationships

As of now in user to user mutual-attention, we only consider influence in terms of attention between each pair of users present in the propagation path individually, which only models relationship between two users at a time regardless of the presence of other users in the sequence. In a real world social network scenario, in some of the cases users only trust and subsequently retweet the content if and only if some particular combination of users have already posted or retweeted the content in their social network fraternity. We call these scenarios as multi-hop latent relationships, in which influence depends on a group of users rather than a single user. We can not capture cues related to such multi-hop latent relationships with the first order user to user mutual-attention described earlier. We propose a novel higher order mutual-attention progression method to deal with it.

3.7. Higher Order mutual-attention Progression

The proposed Higher Order attention progression method is a novel theoretical contribution in the neural attention domain. The intuition behind mutual-attention progression is fairly simple. In the equation 9, values in the score matrix $\underset{m\times m}{\mathrm{S}}$ represent the direct influence relationships between each possible user pairs in the propagation path. Now let's consider a matrix $\underset{m\times m}{\mathrm{S}^2}$ which is computed as:

$$\underset{m\times m}{\mathrm{S}^2} = \underset{m\times m}{\mathrm{S}} \times \underset{m\times m}{\mathrm{S}} \tag{12}$$

Each value in the matrix S^2 represents the indirect influence or affinity between two given users in the propagation path sequence.

$$S^2_{i,j} = \sum_k S_{i,k} \times S_{k,j} \tag{13}$$

This represents the influence between pair of users i and j, encompassing all other users. In the similar fashion we can compute more higher order influence matrices.

$$\begin{aligned} \underset{m\times m}{\mathrm{S}^3} &= \underset{m\times m}{\mathrm{S}^2} \times \underset{m\times m}{\mathrm{S}} \\ \underset{m\times m}{\mathrm{S}^4} &= \underset{m\times m}{\mathrm{S}^3} \times \underset{m\times m}{\mathrm{S}} \end{aligned} \tag{14}$$

Now to compute attention scores, we use row-wise max pooling similar to equation 4 as:

$$\begin{aligned} A'_{Co} &= Softmax(\underset{row}{max} S^2) \\ A''_{Co} &= Softmax(\underset{row}{max} S^3) \\ A'''_{Co} &= Softmax(\underset{row}{max} S^4) \end{aligned} \tag{15}$$

Where A'_{Co}, A''_{Co} and A'''_{Co} are the learned attention weight vectors from second order, third order and fourth order of mutual-attention progression. Finally, higher order user to user mutuallly-attended representations R'_C, R''_C, R'''_C etc can be computed as:

$$\begin{aligned} R'_A &= \sum_{\boldsymbol{i}=1}^{m} f(u_{\boldsymbol{i})A'_{Co\boldsymbol{i}}} \\ R''_A &= \sum_{\boldsymbol{i}=1}^{m} f(u_{\boldsymbol{i})A''_{Co\boldsymbol{i}}} \\ R'''_A &= \sum_{\boldsymbol{i}=1}^{m} f(u_{\boldsymbol{i})A'''_{Co\boldsymbol{i}}} \end{aligned} \tag{16}$$

3.8. Prediction Layer

At the prediction stage, we have two kinds of representations of the propagation path sequence, a representation encoded by LSTM based encoder as R_C and the representations computed using higher order mutual-attention progression as R'_A, R''_A, R'''_A, R''''_A and so on, representing first order, second order, third order and fourth order mutual-attention representations. We compute the cumulative representation from the all higher order mutual-attention progression representations by concatenating them.

$$R^f{}_A = R'_A \parallel R''_A \parallel R'''_A \parallel R'''_A \tag{17}$$

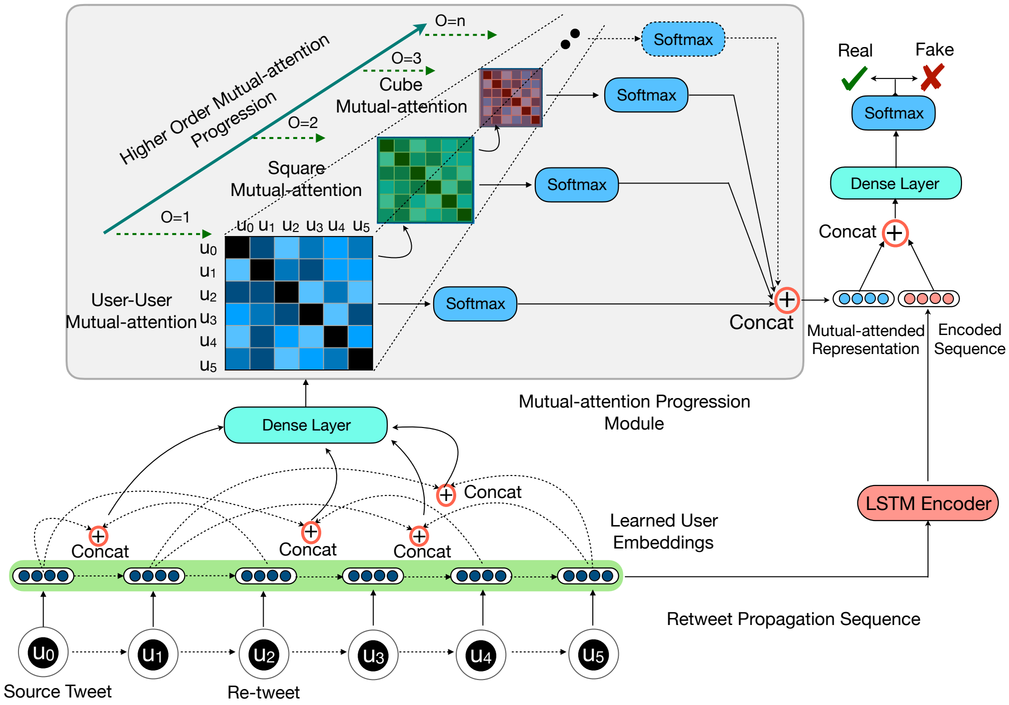


Figure 2. high-level architecture of HiMaP with Higher Order Mutual-attention Progression

We learn a joint representation of $R^f{}_A$ and R_C using a non linear transformation layer.

$$R = ReLU(W_p([R^f{}_A, R_C] + b_p) \quad (18)$$

At the end, we use a Softmax layer for the classification.

$$\hat{y} = Softmax(W_{cl}R + b_{cl}) \quad (19)$$

3.9. Optimization

We use standard Softmax cross-entropy with logits as loss function to train our model.

$$L = -\sum\nolimits_{i=1}^{m} \log \frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum\limits_{j=1}^{n} e^{w_j^T x_i + b_j}} \quad (20)$$

where L is the cost function to be minimized, y_i is class label of x_i.

4. Experimental Setup

4.1. Research Questions

We conduct the experiments with objective to find answers to following research questions:

1. **RQ1**: Is the proposed user to user mutual-attention mechanism useful for the fake news classification?

2. **RQ2**: Does the higher order mutual-attention progression provide useful new and uncovered cues or patterns?

3. **RQ3**: Does the proposed models outperform the state of the art models?

4.2. Datasets

We use two publicly available twitter datasets[1] [16] called Twitter15 and Twitter16 for the evaluation. Twitter15 dataset contains 1490 news stories and Twitter16 dataset

[1] https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0

Table 1. Dataset Statistics

Statistics	Twitter15	Twitter16
News items	1490	818
True news	374	205
fakenews	370	205
Unverified	374	203
Debunking	372	205
Users	276663	173487
Posts	331612	204820
Followers	359385237	225359613
Followings	398394720	249821280

Table 2. Comparison of proposed model with various state of the art baseline models for twitter15 and twitter16 datasets. HiMaP-HO is statistically significant ($p - value = 2.75e^{-3}, 2.03e^{-4}$ for Twitter15 and Twitter16 using pairwise student's t-test)

Twitter15					
Model	**Acc.**	**T F1**	**F F1**	**U F1**	**D F1**
DTC	0.442	0.731	0.351	0.320	0.423
SVM-RBF	0.326	0.442	0.048	0.241	0.273
SVM-TS	0.548	0.773	0.488	0.403	0.479
GRU-RNN	0.641	0.684	0.634	0.688	0.571
TD-RvNN	0.723	0.682	0.758	0.821	0.654
PPC-RNN+CNN	0.842	0.811	0.875	0.790	0.818
HiMaP-FO	0.863	0.822	0.901	0.814	0.826
HiMaP-HO	0.869	0.831	0.889	**0.835**	0.828
HiMaP-HO+Text	**0.880**	**0.837**	**0.917**	0.834	**0.830**

Twitter16					
Model	**Acc.**	**T F1**	**F F1**	**U F1**	**D F1**
DTC	0.462	0.742	0.335	0.337	0.434
SVM-RBF	0.331	0.442	0.085	0.251	0.219
SVM-TS	0.572	0.809	0.469	0.421	0.494
GRU-RNN	0.649	0.691	0.628	0.719	0.592
TD-RvNN	0.743	0.705	0.772	0.842	0.671
PPC-RNN+CNN	0.863	0.826	0.883	0.810	0.824
HiMaP-FO	0.882	0.842	0.936	0.832	0.843
HiMaP-HO	0.890	0.844	0.921	**0.858**	**0.857**
HiMaP-HO+Text	**0.913**	**0.849**	**0.939**	0.854	0.854

Table 3. Performance of mutual-attention Progression method with higher orders

Twitter15	
Model	**Acc.**
HiMaP-FO (O=1)	0.8631
HiMaP-HO (O=2)	0.8663
HiMaP-HO (O=3)	0.8696
HiMaP-HO (O=4)	0.8696
HiMaP-HO (O=5)	**0.8697**

Twitter16	
Model	**Acc.**
HiMaP-FO (O=1)	0.8828
HiMaP-HO (O=2)	0.8891
HiMaP-HO (O=3)	0.8901
HiMaP-HO (O=4)	**0.8908**
HiMaP-HO (O=5)	**0.8908**

contains 818 news stories. In table 1, some statistics related to datasets are shown, for more details of the dataset statistics please refer to [16]. We use Twitter API[2], to crawl the user followers and following as these are not present in datasets.

4.3. Baselines and variants of proposed model

We compare the proposed model with several baseline and state of the art works.

Table 4. Performance of HiMaP with different node embedding methods

Twitter15	
Model	**Acc.**
DeepWalk	0.825
Node2Vec	0.846
APP	0.861
Line	**0.869**

Twitter16	
Model	**Acc.**
DeepWalk	0.850
Node2Vec	0.867
APP	0.889
Line	**0.890**

- **DTC:** [4] This work uses hand crafted text and other statistical features with a decision tree classifier to asses credibility of tweets.
- **SVM-RBF:**[33] This work uses a radial basis function kernel based SVM model to classify news as rumor or non-rumor.
- **SVM-TS:**[15] In this paper, authors create a time series of news characteristics and classify using a SVM model.
- **GRU-RNN:**[14] A gated recurrent unit based model which learns propositional representation of rumors and non-rumors.
- **TD-RvNN:**[17] This work utilizes tree-structured neural networks for rumor representation learning and classification.
- **PPC-RNN+CNN:**[13] In this method, authors propose a multivariate time series representation of news propagation and use combination of GRU and CNN models for classification.

We compare results of above mentioned models with three variants of our HiMaP model.

- **HiMaP-FO:** This is the HiMaP model with first order mutual-attention, where order $O = 1$.
- **HiMaP-HO:** This is the HiMaP model with higher order mutual-attention, where order $O \geq 2$.
- **HiMaP-HO+Text:** This is the HiMaP model with higher order mutual-attention, where order $O \geq 2$ and we also use an LSTM sequence encoder to encode original news text along with propagation path sequence.

4.4. HiMaP Implementation

We use TensorFlow framework to implement our proposed models. We compute overall accuracy and per class F1 scores as performance metrics for evaluation and comparison with the state of the art methods. We use softmax cross entropy with logits as the loss function, learning rate of 0.003 and size of hidden states LSTM units are kept as 100. We tune all the parameters using random search. We

[2] `https://dev.twitter.com/rest/public`

use 50 epochs for each model and use dropout regularization ($keepprob = 0.2$) and early stopping if validation loss does not change for more than 10 epochs. We observe that the sequence length of 35 gives the optimal performance in both the twitter datasets.

The user embeddings are learned using various node embeddings methods namely, DeepWalk [20], Node2vec [8], Line [28] and APP[36]. For all the node embedding methods, we use prescribed parameters and embdedding size as 100. From the retweet networks (trees), we extract all the unique edges and nodes. We assign the weight for each edge as the number of times it occurs in our network. Similarly in case of follower network, we extract all the unique edges and nodes and use node embedding methods to train the node embeddings for each node involved in the network. In case of HiMaP-FO+Text model, we use pretrained GloVe embeddings of 100 dimensions as word embeddings.

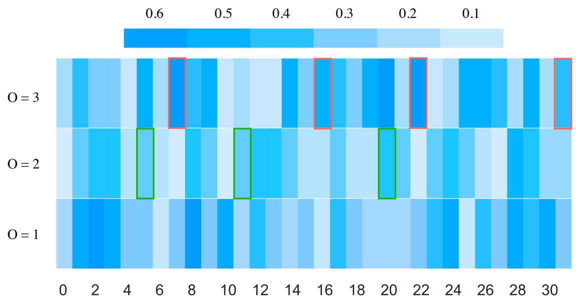


Figure 3. Normalized mutual-attention weight visualization

5. Experimental Results and Analysis

In this section, we evaluate the proposed model and analyse the significance of attention mechanism.

5.1. Results for Twitter15 and Twitter16 datasets

In Table 2, we present the comparison of performance of the proposed model with several baselines and state-of-the-art methods. We can observe that even the basic HiMaP model (HiMaP-FO, where $O = 1$) outperforms all the baselines and state of the art models. Among the baseline methods, we notice that RNN based methods are more effective than other methods. An intuitive explanation for this trend can be the capability of RNN models to easily learn the compositional aspects of news content in GRU-RNN and news propagation sequence in PPC-RNN+CNN and TD-RvNN, without any or with minimal feature engineering. On the other hand, PPC-RNN+CNN outperforms TD-RvNN as they use CNN to capture the local variations within a propagation sequence.

The basic HiMaP model (HiMaP-FO, where $O = 1$) performs better than both the state of the arts (PPC-RNN+CNN and TD-RvNN) as it not only uses LSTM based sequence encoder to capture the compositional aspects but also utilizes user-user mutuallly-attended representations of propagation path, which inherently holds the latent cues and patterns pertaining to influence relationships among users. Therefore we can conclude that research questions **RQ1** and **RQ3** are satisfied. The HiMaP-HO (where $O \geq 2$) model outperforms HiMaP-FO model as it uses mutual-attention progression of higher order, which captures indirect relationships among all pairs of users present in the propagation path sequence.

The HiMaP-HO+Text model uses original news text also with propagation path, which provides additional topical and semantic cues related news text and outperforms all the other models.

5.2. Analysis of HiMaP with Higher Order mutual-attention

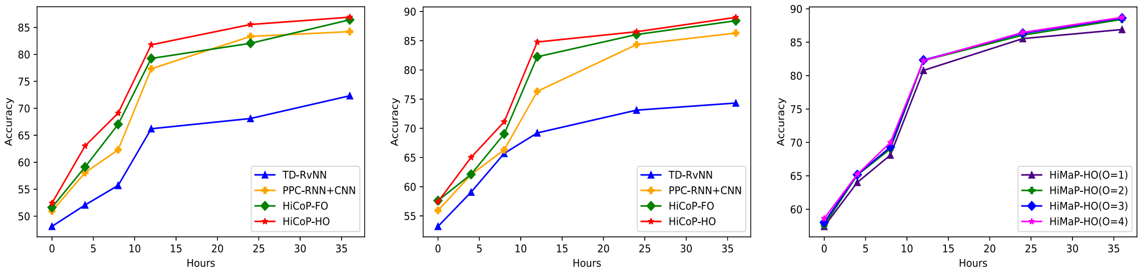
In table 3 we show comparison of HiMaP model with different values of mutual-attention order O. We can observe that performance of HiMaP in terms of accuracy improves with increase in mutual-attention order, this means research question **RQ2** is answered partially as we know now that higher order mutual-attention is useful. However, we notice that after $O = 3$, accuracy starts to saturate for higher orders in both the datasets, therefore we can tune the parameter O and omit computation of higher order mutual-attention. In table 2, where we compare HiMaP with baselines and state of the art models, we use HiMaP results with third order of mutual-attention progression where $O = 3$.

5.3. Analysis of HiMaP with different node embedding methods

In table 4, we show the effect of using different node embedding methods to learn user embeddings. We observe that the Line method outperforms all the other node embedding methods in both the datasets. In table 2, where we compare HiMaP with baselines and state of the art models, we use HiMaP results with Line embeddings. The reason behind the better performance of the Line method can be the suitability of the Line method for graphs with low clustering coefficient and transitivity and we observe for both the Twitter datasets (twitter15 and twitter16), the values of clustering coefficient and transitivity are low.

5.4. Comparison of Early Detection Accuracy

In Figure 4, we compare the early detection performance of the proposed models with the state-of-the-art models. We plot the overall accuracy vs elapsed time since the original



(a) In case of Twitter15 dataset

(b) In case of Twitter16 dataset

(c) HiMaP with different orders

Figure 4. Comparison of HiMaP models with state-of-the-art models in terms of early detection accuracy at different timestamps of the news propagation, depicted in plots (a) and (c). Comparison of HiMaP models with different mutual-attention levels in terms of early detection accuracy at different timestamps of the news propagation, depicted in plot (c).

tweet is posted. We can observe in Figure $4(a)$ and $4(b)$ that for both the twitter15 and twitter16 datasets, HiMaP-HO outperforms state-of-the-art models at each time step. The better performance of HiMaP can be attributed to the learning of additional and useful propagation patterns due to higher-order mutual-attention progression method. In Figure 4, we also compare the early detection performance of HiMaP models with different values of order O. We can observe in $4(c)$ that there is significant improvement form $O = 2$ to $O = 3$ but there is not much significant improvement above third order ($O \geq 3$).

5.5. Mutual-attention Visualization and Analysis

In this section, we explain the visualization of attention weights from three levels of mutual-attention progression, for a propagation sequence of a anecdotal news example. In figure 3, there are 32 users in the propagation path of news item. Each strip in figure 3, represents a different order of mutual-attention, first strip is the depiction of attention weights from first order mutual-attention, where $O = 1$ and similarly second and third strip depict weights from second and third order mutual-attention weights for the same propagation sequence. The depth of the colors in the rectangles in each strip represents the distribution of attention weights among users present in propagation path. We do not reveal the identity of users for sake of privacy and twitter's policy. We observe that in the first order mutual-attention, users with high number of followers get more attention weights. In contrast to first order mutual-attention, in the second and third order, some of the users with less followers and prestige also get higher attention weights (highlighted rectangles with green and red borders). We also notice that beyond third order mutual-attention $O = 3$, there are no significant changes in the attention pattern. We conclude that higher order mutual-attention captures new, uncovered and significant latent patterns and hence research question **RQ2** is satisfied.

6. Conclusions

In this paper, we propose a novel user to user mutual-attention progression method to model influence relationships among users, present on news propagation path to detect fake news. This method allows us to capture both the direct and indirect (multi-hop) relationships between each pair of users. Experiment with two publicly available twitter datasets, shows the effectiveness of our model, compared to state-of-the-art models, in terms of early detection and overall accuracy. We also notice that higher-order mutual-attention progression method captures useful new, uncovered patterns and provides the classifier with the cues pertaining to propagation of true or fake news. The proposed attention progression trick can also be useful in other application scenarios such as in case of word to word attention in sentences.

In future, we plan to conduct an experiment, related to evidence extraction, using mutual-attention weights from different levels of higher order mutual-attention. Effectiveness of the Higher-order Attention trick can also be utilized with recent transformer architectures.

References

[1] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: Quantifying influence on twitter. WSDM '11, page 65–74, New York, NY, USA, 2011.

[2] Sambaran Bandyopadhyay, Ramasuri Narayanam, and M. Narasimha Murty. A generic axiomatic characterization for measuring influence in social networks. pages 2606–2611, 2018.

[3] Yoshua Bengio and Yann LeCun, editors. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[4] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *WWW*, 2011.

[5] Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, and Krishna P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2010.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. KDD '16, page 855–864, 2016.

[9] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *SocInfo*, pages 228–243, 2014.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[11] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[12] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. Rumor detection over varying time windows. volume 12, page e0168344, 01 2017.

[13] Yang Liu and Yi-fang Brook Wu. Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks. *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 354–361, 2018.

[14] 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. IJCAI'16, page 3818–3824, 2016.

[15] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. CIKM '15, page 1751–1754, 2015.

[16] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors in microblog posts using propagation structure via kernel learning. ACL'17, pages 708–717.

[17] Jing Ma, Wei Gao, and Kam-Fai Wong. Rumor detection on twitter with tree-structured recursive neural networks. pages 1980–1989, July 2018.

[18] Rahul Mishra and Vinay Setty. Sadhan: Hierarchical attention networks to learn latent aspect embeddings for fake news detection. ICTIR '19, page 197–204, 2019.

[19] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A Decomposable Attention Model for Natural Language Inference. 2016.

[20] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. KDD '14, page 701–710, 2014.

[21] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *WWW*, pages 1003–1012, 2017.

[22] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. Declare: Debunking fake news and false claims using evidence-aware deep learning. In *EMNLP*, pages 22–32, 2018.

[23] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. EMNLP '11, 2011.

[24] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. Neural user response generator: Fake news detection with collective user intelligence. IJCAI '18, pages 3834–3840, 2018.

[25] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *EMNLP*, pages 2931–2937, 2017.

[26] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806. ACM, 2017.

[27] Kai Shu, Suhang Wang, and Huan Liu. Beyond news contents: The role of social context for fake news detection. In *WSDM*, pages 312–320. ACM, 2019.

[28] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. WWW '15, 2015.

[29] Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. Reasoning with sarcasm by reading in-between. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1010–1020, 2018.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. abs/1706.03762, 2017.

[31] Liang Wu and Huan Liu. Tracing fake-news footprints: Characterizing social media messages by how they propagate. WSDM '18, 2018.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2048–2057. PMLR, 2015.

[33] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. Automatic detection of rumor on sina weibo. MDS '12, 2012.

[34] Shuo Yang, Kai Shu, Suhang Wang, Renjie Gu, Fan Wu, and Huan Liu. Unsupervised fake news detection on social media: A generative approach. 02 2019.

[35] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[36] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. AAAI'17, 2017.