Modeling Content Interaction in Information Diffusion with Pre-trained Sentence Embedding

Qinyuan Ye*, Yuejiang Li*, Yan Chen†, H. Vicky Zhao*
*Dept. of Automation and Inst. for Artificial Intelligence, Tsinghua Univ.
Beijing National Center for Info. Science and Technology (BNRist), P. R. China
†School of Info. & Comm. Engr., Univ. of Electronic Science and Technology of China, P. R. China

***Abstract*—Social networks have become indispensable parts of our daily life, and therefore understanding the process of information diffusion over social networks is a meaningful research topic. Usually, multiple pieces of information do not spread in isolation; rather, they interact with each other throughout the diffusion process. This paper aims to quantify these interactions by modeling users' forwarding behavior after reading a series of information. Inspired by several successful components prevalent in recent research of deep learning, *i.e.*, long short term memory (LSTM) network and bi-directional encoder representation from transformers (BERT), we designed IMM Enhanced model and InfoLSTM model. In our experiments on real-world Weibo dataset, both models significantly outperform baselines such as the prior IMM model and IP model, with IMM Enhanced model improving 23.52% and InfoLSTM model improving 32.56% in F1 score (absolute value) compared to that of baseline IMM model. In addition, we visualize the dataset and the parameters learned in IMM Enhanced model, which further enables us to discuss the relationship between text similarity and information diffusion interaction with case studies.**

***Keywords*—social network, information diffusion, forward, pre-trained model, visualization.**

I. Introduction

The invention of social network connects people all over the world tightly together. Every day, countless new information is posted online and forwarded from one user to another. According to 2018Q4 financial statements, the number of monthly active users of Twitter reaches 126 million. For Weibo, Twitter's counterpart in China, the number is 200 million. With a large-scale user population like this, the impact of information diffusion over social network is non-negligible, and research on such diffusion process is necessary.

There are many interesting phenomena in social network. For example, a severe earthquake hit Japan in March 2011, and it further triggers tsunami and nuclear leakage. These news events went viral on social networks. Meanwhile in China, another topic, buying iodized salt, became trending. It was falsely advertised that eating iodized salt can prevent being affected by nuclear radiation, and lots of people believe in this story. 'Salt', 'earthquake' and 'tsunami' seemed to be unrelated at first, but we can still find logical explanations for this whole event. Another pair of events was the detention of former Canadian diplomat Michael Kovrig by Chinese government and the detention of Wanzhou Meng, CFO of Huawei Inc., by Canadian government. The two events almost happened simultaneously and both landed on the trending charts of Weibo in December 2018. Imagine a Weibo user who is interested in politics, he may be attracted by both events and forward related microblogs. In the two examples mentioned above, we found that the information diffusion processes over social network are not isolated. From a micro-level perspective, the forwarding decision may be influenced by what a user have seen previously; from a macro-level perspective, it appears as multiple sources of information interact with each other during the diffusion process, and the interactions further influence the diffusion speed, scale and final state.

Studying these interactions are of great importance. For users in social network, understanding the process helps increase their personal impact. Also, for enterprises, it helps promote products or control unreal rumors. In this paper, we focus on the problem of quantifying content interactions in information diffusion by modeling users' forwarding behavior after they read a series of information.

In recent years, with the rapid development of machine learning, and their applications in computer vision, natural language process, speech recognition, etc., many sophisticated tasks are perfectly accomplished by algorithms at human-intelligence level. A large proportion of information on social network is in the form of text, and thus introducing methods successful in natural language processing may be helpful to understanding content interaction. Also, deep neural networks and recurrent neural networks prove to be effective in real-world prediction tasks. Introducing these models can help improve the accuracy of predictions on user forwarding behavior. In this paper, we attempt to introduce these latest advanced into the problem that we focus on.

In particular, we modified IMM model proposed in [1] by introducing BERT pre-trained model [2] as text encoder. Also, we proposed a new model, InfoLSTM, which incorporate BERT pre-trained models and LSTM network. Both models significantly outperform baselines models. In our experiments on real-world datasets, InfoLSTM achieves 68.77% in AUC and 64.84% in F1 score. Meanwhile, IMM enhanced model achieves 52.25% in AUC and 55.82% in F1 score. With the baseline model – IMM model, these metrics are 26.86% and 32.28% respectively.

IMM model explicitly describes pair-wise interactions from

This work is supported by the National Key Research and Development Program of China (2017YFB1400100)

a probabilistic perspective and some of the parameters learned are interpretable. Our IMM enhanced model keeps this advantage. We leverage visualization tools to empirically analyze the dataset content and parameters learned by IMM enhanced model. We further discuss the relationship between text similarity and content interaction.

The remaining parts of the paper is organized as follows. In section 2, we introduce related works, including recent research in information diffusion, recurrent neural networks and sentence embedding. In section 3, we introduce our experiment setups in detail, including dataset generation and baseline model introduction. We introduce IMM enhanced model and InfoLSTM in section 4 and discuss their results in section 5. Furthermore, visualization of IMM enhanced model is shown in section 5, with in-depth case study on two example microblogs. Conclusion is provided in section 6.

II. Related Works

A. Information Diffusion over Social Network

Current study of information diffusion can be generally categorized into two groups: model-based methods and data-driven methods. With model-based methods, some researchers compare information to epidemic, and simulate the diffusion process with epidemic model [3]. Meanwhile, evolutionary game theory, which was originally designed for modeling evolution in eco-system, is customized and applied to information diffusion [4], [5]. Players (social network users) adopt different strategies (forward or not forward) to achieve maximum payoff in the graph (social network). The diffusion dynamics and final stable state can also be described with graphical evolutionary game theory.

With data-driven methods, usually a statistical model is learned with a train set, and predictions are done on test set to evaluate model performance. IMM model [1] is designed to model the probability of a user forwarding the contagion (*i.e.* microblog) he is currently viewing, given a sequence of contagions he has previously read. Some empirical analyses is done with IMM model, and the results provide compelling hypotheses for the principles of interactions. Later on, a unified framework – Interaction-aware diffusion (IAD) [6] is proposed, in which the idea of 'interactions' is extended into three forms: (1) contagion-contagion interactions; (2) user-contagion interactions; (3) user-user interactions. The performance of prediction task is improved with IAD framework.

B. Recurrent Neural Network and LSTM

Recurrent neural networks (RNNs) are proved to be successful in prediction tasks with sequential inputs. However, their ability to acquire useful information in long sequences is not desirable. In order to overcome this vulnerability, Hochreiter et al. [7] proposed long short term memory (LSTM) network with memory mechanism, which improve the network's ability in dealing with long sequences. Specifically, LSTM maintains a cell state C_t at time t, and decides which parts of C_t is preserved at net time stamp $t+1$. It is suitable to adopt LSTM network for modeling human behavior, especially for modeling forwarding behavior given a sequence of previously-seen microblogs as input. LSTM may simulate users' forgetting and remembering process in browsing microblogs, and make decisions based on the memories left.

C. Sentence Embedding and BERT

The concept of embedding originates from word embedding, where each word is mapped to a low-dimensional vector. The characteristics of each word are preserved in latent space by aligning similarity in corpus with vector dot product similarity in latent space. For example, it is found that the vector difference between uncle and aunt is similar to that of sir and madam. Models generating embeddings are called pre-trained models, since these embeddings are usually trained ahead of time and then applied to downstream tasks. Popular word-level pre-trained models include word2vec [8] and GloVe [9].

Word embedding achieves huge success, but still fails to extract semantics and logics at sentence or paragraph level. Recently, researchers start to focus on pre-training at higher level. Some successful attempts include Embedding from Language Models (ELMo) [10] proposed by AllenNLP and GPT model [11] proposed by OpenAI.

Bidirectional Encoder Representation form Transformers (BERT) is the latest and novel language representation model proposed by Google AI in October, 2018 [2]. It brought natural language processing (NLP) to a new era by refreshing the state-of-the-art in 11 NLP tasks. One of BERT's application is to generate sentence embedding for downstream tasks. BERT is trained with two tasks: (1) masked language modeling, *i.e.*, fill in the blanks in one sentence; (2) next sentence prediction, *i.e.*, determine whether one sentence is the next sentence of another. In summary, BERT model generates a vector representation for each sentence input. This vector maintains the semantics of the sentence and can be used in prediction tasks.

III. Experiment Setup

A. Problem Definition

Here we formally define the problem we study in this paper. The basic unit of 'information' is one original microblog (*i.e.*, contagion). The scenario that we're modeling is that some user view the following microblogs sequentially:

$$\{Y_k = u_k, ..., Y_2 = u_2, Y_1 = u_1, X = u_0\},$$

and we predict the probability of this user finally forwarding microblog $X = u_0$, denoting this probability as

$$P(X|\{Y_k\})_{k=1}^{K}$$

We would like to clarify that, though different forms of interactions have already been introduced in previous work, in this paper we only focus on content interactions (*i.e.*, contagion-contagion interactions in [6]) and thus we choose not to include user-specific information. Still, we believe user profile is useful and leave this as future work.

B. Dataset and Pre-processing

We adopt Sina Weibo dataset[1], which was originally collected for studying social influence locality [12] (*i.e.*, how user forwarding behavior is influenced by his neighbors). The dataset contains a comparatively full picture of all forwarding activities on Sina Weibo in July, 2012. Also the static user following network is included. The basic information of this dataset is shown in Table I.

TABLE I: Summary of Sina Weibo Dataset

Source	#users	#following	#microblog	#forwarding activity
Sina Weibo	1776950	308489739	300000	23755810

As the dataset was originally collected for another purpose, we need to parse the dataset into the form of our target problem. The major challenge is transforming the raw data into the format of the information stream that user actually sees. This step requires enumerating each follower of each user who forwards the microblog of interest. It also requires large-scale file I/O. This step is fulfilled with multi-process programs. Table II shows the snippet of information stream that an exmaple user sees.

TABLE II: Example of Information Stream

username	weiboID	timestamp	type	info
981713	3462659405529840	2012-07-01-01:45:22	1	
1466971	3462378476160530	2012-07-01-08:16:00	1	
475730	3424653030211020	2012-07-01-08:26:25	1	
85053	3462908140621280	2012-07-01-09:11:41	1	Original
896	3462911777809630	2012-07-01-10:20:09	0	
896	3462911777809630	2012-07-01-10:21:38	0	
218702	3462931406536090	2012-07-01-12:02:06	1	
968432	3462775295099230	2012-07-01-14:20:59	1	

Following [1], we generate instances of sequence $\{Y_k, ..., Y_2, Y_1, X\}$ and label $y \in \{0, 1\}$, where 1 indicates that X was forwarded and 0 indicates that it is was not forwarded. In reality, the number of negative samples significantly exceed that of positive samples. To avoid bias in model training, we keep all positive instances and maintain the positive : negative rate to be $1 : 5$ during dataset generation. Due to the abundance of raw data, we only keep Weibo activities in the range of July 1st to July 5th. $\tilde{1}$70k instances were generated with the above mentioned process, and they were further split into train set, dev set and test set with the proportion of 80%, 10% and 10%.

C. Baseline Models

a) IP Model: Infection Probability (IP) model assumes the diffusion process of each contagion is independent, *i.e.*, users have no memory when they decide whether to forward or not:

$$P(X = u_i | \{Y_k\}_{k=1}^K) = P(X = u_i) \tag{1}$$

in which $P(X = u_i)$ is the number of times of u_i being retweeted divided by the number of times of being seen:

$$P(X = u_i) = \frac{n_{\text{retweet}}}{n_{\text{seen}}} \tag{2}$$

[1]http://www.aminer.cn/influencelocality

IP model is completely based on statistics acquired on train set and does not require any parameter tuning. In our experiments, we calculate $P(X = u_i)$ for each microblog that appears in train set. During evaluation, $P(X = u_i)$ is retrieved from our calculation if the u_i in test instances appears in train set. Otherwise, we set predicted $P(X = u_i)$ to be zero.

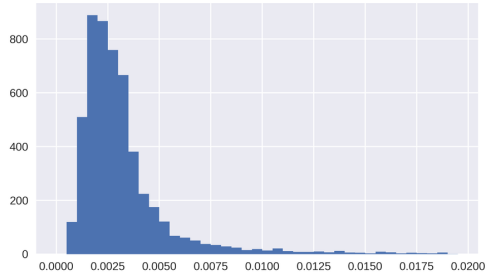


Fig. 1: Histogram of $P(X = u_i)$ in IP Model

Fig. 1 shows the distribution of such $P(X = u_i)$. From the figure, we have the following observations:

- $P(X = u_i)$ of most microblogs are in the interval of $(0, 0.005)$. The 90% percentile is around 0.0055.
- The number of instances decreases as $P(X = u_i)$ increases after $P(X = u_i)$ exceeds 0.005. Popular microblogs are rare over the whole network.
- There is no microblog whose $P(X = u_i)$ is equal to zero. This is due to data collecting settings. This issue potentially creates bias in the trained model.

b) IMM Model: IMM model was originally proposed in [1]. The main idea is to decompose the conditional probability into several independent terms under certain assumptions.

$$P(X | \{Y_k\}_{k=1}^K) = \frac{1}{P(X)^{K-1}} \prod_{k=1}^{K} P(X|Y_k) \tag{3}$$

In addition, the interactions between information is described with an additive term $\Delta_{cont.}(u_i, u_j)$.

$$P(X = u_j | Y_k = u_i) \approx P(X = u_j) + \Delta_{cont.}(u_i, u_j) \tag{4}$$

The $P(X = u_j)$ term in Eq. (4) is the same as in Eq. (2) in IP model.

Suppose we have n microblogs of interest, calculating $\Delta_{cont.}(u_i, u_j)$ for each pair of u_i and u_j results in $n \times n$ parameters and is thus impractical due to the number of learnable parameters. IMM model uses clustering technique by describing each microblog u_i with a T-dimensional vector $\mathbf{M}_i$ which fits $\sum_{t=1}^{T} M_{i,t} = 1$. Then $\Delta_{cont.}(u_i, u_j)$ is calculated in a bi-linear form described in Eq. (5).

$$\Delta_{cont.}(u_i, u_j) = \sum_t \sum_s M_{j,t} \Delta_{clust}(c_t, c_s) M_{i,s} \tag{5}$$

In this way, the number of learnable parameters is reduced from $n \times n$ to $T \times n + T \times T$. The workflow of IMM model is summarized in Fig. 2.

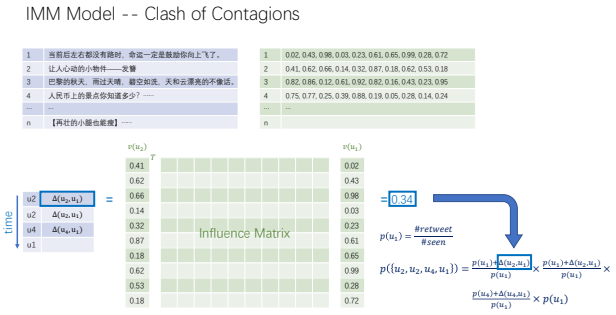


Fig. 2: Workflow of IMM Model

In practice, we found that with random initialization, the final output $P(X|\{Y_k\}_{k=1}^K)$ and intermediate result $P(X = u_j|Y_k = u_i)$ may exceed the probability limit of $[0, 1]$, and leads to wrongful calculation. In order to deal with this challenge, two minor modifications are made in our implementation:

- We use clipping function to make sure $P(X|\{Y_k\}_{k=1}^K)$ and $P(X = u_j|Y_k = u_i)$ remains in the limit of $[0, 1]$. The clipping function $y = c(x)$ is designed as follows:

$$c(x) = \begin{cases} 10^{-6}, & x \leq 0 \\ x, & 0 < x < 1 \\ 1, & x \geq 1 \end{cases} \tag{6}$$

- With clipping function, the gradient may not be calculated when x is out of the limit, and the parameters won't be updated. We need additional steps to make sure x stays in the limit. We design a penalty function $p(x)$ for each $P(X|\{Y_k\}_{k=1}^K)$ or $P(X = u_j|Y_k = u_i)$. This function is arbitrarily chosen to penalize intermediate results out of the $[0, 1]$ interval. Other options may be explored in future work.

$$p(x) = \begin{cases} x^2, & x \leq 0 \\ 0, & 0 < x < 1 \\ (x-1)^2, & x \geq 1 \end{cases} \tag{7}$$

The sum of all penalty terms is added to objective function O_{old}, with the weight of k. k will be tuned as a hyper-parameter.

$$\mathrm{O}_{new} = \mathrm{O}_{old} + k \times \sum_x p(x) \tag{8}$$

D. Implementation Details

In this section we summarize common implementation details for all models, including IMM model and the two models (IMM enhanced model and InfoLSTM model) that will be introduced in next section.

- Stochastic gradient descent (SGD) is adopted for model training. Initial learning rate is a hyper-parameter to be tuned from $\{1, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001\}$.
- Dynamic learning rate reduction is adopted. If the F1 score on dev set was not improved for three consecutive epoches, learning rate will be reduced by half.
- In IMM model, clustering dimension T determines how much information is captured and described by the clustered vector $\mathbf{M}_i$. T is tuned from $\{8, 16, 32, 64, 128, 256\}$.
- In IMM enhanced model, the dimension of hidden layer is also tuned from $\{8, 16, 32, 64, 128, 256\}$.
- In InfoLSTM model, dropout [13] is adopted and the dropout rate is selected from $\{0.5, 0.4, 0.3, 0.2, 0.1, 0\}$.

E. Evaluation

Though the desired output is a float between $[0, 1]$ representing probability, the outputs of different models are generated from different perspectives (shown in Table III) and thus it is not reasonable to set a unified threshold for all models during evaluation.

TABLE III: Comparison of Model Output

Model	Means to Generate Output Prob.
IP Model	Statistical Calculation
IMM / IMM Enhanced Model	Conditional Probablilty
InfoLSTM	Sigmoid Function

For fair comparison among different models, we tune a threshold on a 10% dev set and apply this threshold to test set evaluation. We calculate accuracy, precision, recall and f1 score for comprehensive comparison. In addition, precision-recall curve is plotted and AUC is calculated.

For each model, we train it for 5 times with different random seed. This is to test model's robustness and reliance on random seed. We report average, standard deviation, max and min of all four metrics.

IV. Models

A. Defects of Existing Models

Though IMM model was reported to improve the performance greatly compared to IP model, several defects still exists in when it is applied to real-world situations. Meanwhile, cutting-edge advances in deep learning and natural language processing achieve huge success, and provide new perspectives to improve the performance. In particular,

- During the training process of IMM model, each microblog is treated as an independent unit. The clustered vector $\mathbf{M}_i$ is learned from scratch (randomly initialized vectors), and microblog text is not taken into consideration at all. Apparently, the content of microblog makes significant contributions to our decisions and should be included as input in some reasonable form.
- IMM model can only be applied to microblogs seen in train set. For new microblogs, the clustered vector $\mathbf{M}_i$ cannot be retrieved directly. In reality, new microblogs are constantly posted and we may be interested in their interaction with other microblogs as well. Though in [6] some attempts have been made by leveraging topic modeling method [14], novel sentence embedding models such as BERT can me more expressive and powerful.

- The number of trainable parameters in IMM model is still large. These parameters are 'sparse' since each microblog is mapped to a unique clustered vector. For some microblogs that only appears few times in train set, the corresponding parameters may not be sufficiently trained.
- IMM model only captures pair-wise interactions. In other words, sophisticated logic chain that includes more than two microblogs can not be captured. For example, a user may found u_2 to be a news article, and u_1 to be an opinion piece which he does not agree. When he is deciding whether to forward u_0, an opinion piece that he agrees with, the decision is made upon both u_1 and u_2. Current IMM model fails to extract interactions like this.

Noticing these defects in existings models, we designed InfoLSTM model and IMM Enhanced model, in which LSTM and BERT are included to smooth over these short-comings.

B. InfoLSTM Model

InfoLSTM uses BERT pre-trained model to encode the text of each microblog. It also integrates LSTM and logistic regression into a prediction model. Specifically, InfoLSTM model includes the following components:

a) Input Generation: Each microblog u_i is firstly indexed as i, and then map to a d_b-dimensional vector $\mathbf{v}(u_i)$ with the help of BERT pre-trained model. With the preprocessing steps described in previous section, each instance is formulated into a sequence of microblogs $\{Y_k = u_k, ..., Y_2 = u_2, Y_1 = u_1, X = u_0\}$, and they are further mapped to become $\{\mathbf{v}(u_k), ..., \mathbf{v}(u_2), \mathbf{v}(u_1), \mathbf{v}(u_0)\}$, which formulates a matrix of $\mathbb{R}^{d_b \times (k+1)}$.

b) LSTM layers: We adopt 2 layers of LSTM. We also use dropout [13] with rate p to prevent over-fitting. The hidden dimension of LSTM is denoted as d_q. The final hidden state $\mathbf{q} \in \mathbb{R}^{d_q}$ is used as the output of LSTM layers.

c) Logistic Regression: Based on final hidden state $\mathbf{q}$ calculated with LSTM layers, we predict the user forwarding probability with logistic regression, *i.e.*,

$$p(X|\{Y_k\}_{k=1}^K) = \text{Sig.}(w_0 + w_1 q_1 + w_2 q_2 + ... + w_{d_q} q d_q) \quad (9)$$

in which the Sig. is the Sigmoid function:

$$\text{Sig.}(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

The workflow of InfoLSTM is also summarized in Fig. 3. The hyper-parameters tuned on dev set is logged in Table V in Appendix.

C. IMM Enhanced Model

IMM Enhanced model is based on original IMM model [1], with some several modifications to better suit the BERT pre-trained model. The workflow of IMM Enhanced Model is summarized in Fig. 4. It includes the following components:

a) Input Generation: This step is the same as in InfoLSTM model.

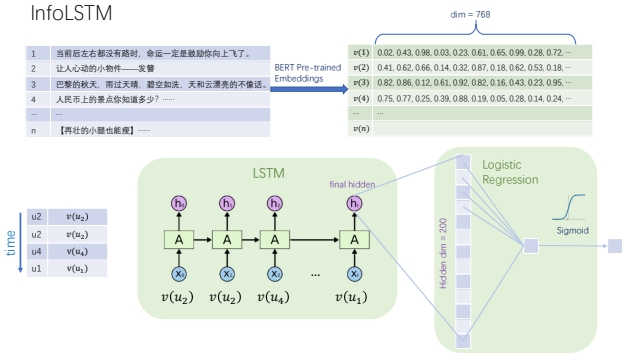


Fig. 3: Workflow of InfoLSTM Model

b) Linear Layer: The default output dimension of BERT model is $d_b = 768$, which is informative but infeasible as it creates a large number of parameters to train (*i.e.* $\mathbf{\Delta}_{clust.} \in \mathbb{R}^{d_b \times d_b}$). We reduce these embeddings to a smaller dimension, d_h, with linear transform. That is, given the embedding vector $\mathbf{v}(u_i) \in \mathbb{R}^{d_b}$, we transform it into $\mathbf{w}(u_i) \in \mathbb{R}^{d_h}$:

$$\mathbf{w}(u_i) = \mathbf{A}\mathbf{v}(u_i) + \mathbf{b} \quad (11)$$

where $\mathbf{A} \in \mathbb{R}^{d_h \times d_b}$ and $\mathbf{b} \in \mathbb{R}^{d_h}$.

c) Calculate Interaction Term: Following the idea in original IMM model, the interaction term $\Delta_{cont.}(u_i, u_j)$ is calculated.

$$\Delta_{cont.}(u_i, u_j) = \mathbf{w}(u_i)^{\mathrm{T}} \mathbf{\Delta}_{clust.} \mathbf{w}(u_j) \quad (12)$$

d) Prediction: With $\Delta_{cont.}(u_i, u_j)$ from previous step, we can easily calculate $P(X|Y_k), \forall k \in \{1, 2, ..., K\}$ following Eq. (4), and then calculate our final output $P(X|\{Y_k\}_{k=1}^K)$ following Eq. (3).

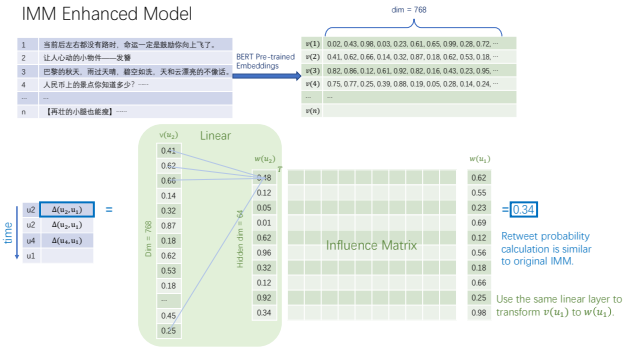


Fig. 4: Workflow of IMM Enhanced Model

V. Results and Analysis

A. Overall Comparison

Results of baseline and proposed models are reported in Table IV. From the results we have the following observations:

- At first glance, IP model has already obtained prediction ability to some degree, achieving F1 score of 28.94%. However, noting that the dataset we generate has a positive:negative ratio of 1 : 5, the model would get a

TABLE IV: Experiment Results

		AUC	Precision	Recall	F1
IP	one run	15.80	16.96	98.35	28.94
IMM	mean ± std	26.86 ± 1.00	37.69 ± 1.83	28.35 ± 2.76	32.28 ± 1.86
	min / max	25.79 / 28.47	35.32 / 40.44	24.86 / 31.37	29.81 / 34.29
IMM Enhanced	mean ± std	52.24 ± 10.40	57.13 ± 1.75	55.77 ± 12.98	55.81 ± 7.61
	min / max	38.71 / 62.23	54.01 / 58.18	36.09 / 66.22	44.50 / 61.74
IMM Enhanced	mean ± std	50.30 ± 9.23	55.74 ± 2.34	57.02 ± 8.31	56.17 ± 4.86
(fix embedding)	min / max	41.65 / 62.24	53.22 / 58.50	45.55 / 65.55	50.44 / 61.83
InfoLSTM	mean ± std	68.77 ± 0.07	61.49 ± 0.47	68.59 ± 0.50	64.84 ± 0.09
	min / max	68.67 / 68.83	60.85 / 62.10	68.05 / 69.24	64.75 / 64.98
InfoLSTM	mean ± std	68.77 ± 0.07	61.44 ± 0.78	68.66 ± 0.96	64.84 ± 0.09
(fix embedding)	min / max	68.66 / 68.83	60.22 / 62.15	67.55 / 70.11	64.74 / 64.98

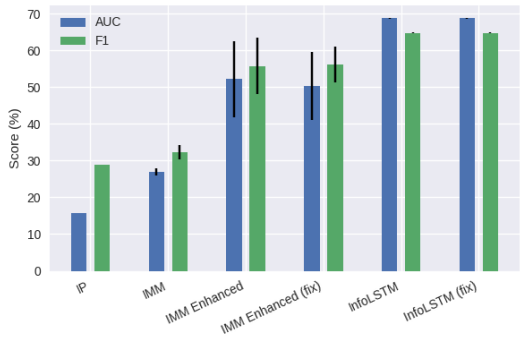


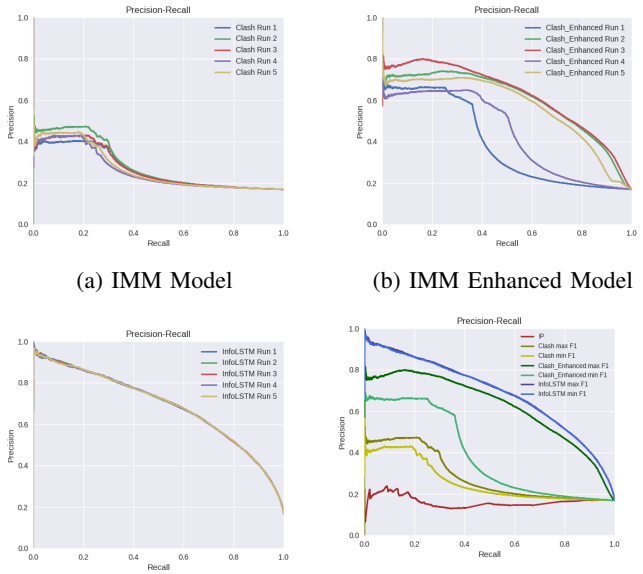
Fig. 5: Performance Comparison of Four Models

28.57% F1 score for simply predicting TRUE for every instance. In fact, the power of IP model is very limited.

- IMM model achieves the average F1 score of 32.28%, which is better than IP model. AUC is also improved. Meanwhile, the precision and recall is more balanced (*i.e.*, precision is close to recall). However, the standard deviation of the 5 runs is large – due to the influence only brought by random seed, there is a 5% absolute-value gap between maximum and minimum F1 score.
- IMM Enhanced model outperforms IMM model in all four metrics. The minimum F1 score of IMM enhanced model exceed the maximum F1 score of IMM model. It was widely acknowledge that using pre-training model helps the model converge to a better optimum. This argument is further validated with the comparsion between IMM enhanced model and IMM model. However, the large standard deviation problem becomes more severe with IMM Enhanced model. The standard deviation of F1 score reaches 7.61%, and the std of AUC reaches a surprising 10.40%.
- Performance of InfoLSTM model is further improved compared to other models. The F1 score already reaches a comparatively high 64.75%. In addition, the standard deviation of InfoLSTM model is significantly smaller than that of IMM model or IMM enhanced model. InfoLSTM beats IMM enhanced model in all aspects when it comes evaluation metrics. However, IMM enhanced model is superior in its interpretability.
- As BERT pre-trained model is introduced in IMM enhanced model and InfoLSTM model, we try training these models under two different settings: pre-trained embeddings (1) are fixed; (2) are trainable. For InfoLSTM model, the difference is marginal. For IMM enhanced model, standard deviation is smaller when embeddings are fixed.

B. *Precision-Recall Curves*

Precision-Recall (PR) Curves are summarized in Fig. 6. From the curves we have the following observations:



(a) IMM Model (b) IMM Enhanced Model

(c) InfoLSTM Model (d) Overall Comparison (with Max and Min)

Fig. 6: Comparison of Precision-Recall Curves

- From Fig. 6 (a) we learn that the PR curves of IMM model are consistent over 5 runs, but are still slightly influenced by different random seeds. Precision maintains at 0.4 when Recall < 0.2, and starts to decrease when Recall > 0.2.
- From Fig. 6 (b) we find that three of the curves are similar (#2, #3, #5), while the other two curves starts to drop significantly when Recall ≈ 0.5. The influence of random seeds is more severe with IMM Enhanced model.
- From Fig. 6 (c) we learn that the five runs of InfoLSTM are very consistent and are barely influenced by random seeds. Also, the trade-off between precision and recall is smoother.

In order to comprehensively compare different models, we plot the best and worst run (based on F1 score) of the three models together in Fig. 6 (d), and we find that:

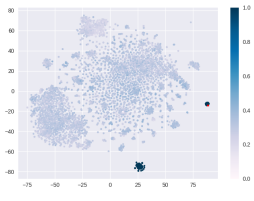
- All curves converge to bottom-right corner of Recall $= 1$ and Precision ≈ 0.17. This is reasonable since we adopt the positive-negative ratio of $1:5$ when generating the dataset. When the threshold is set to 0, all test cases will be predicted to be positive (*i.e.*, the user will forward), and yielding the $(1, \frac{1}{6})$ in PR curve.
- Usually precision and recall have a trade-off relationship, and PR curves appear to be a line from top-left to bottom-right. However, with IP model, the curve is unstable and goes up and down when Recall $\in (0.2, 1.0)$. This observation verifies that the power of IP model is limited.

- The conclusion of performance comparison is the same as discussed in Section V-A. That is, the worst run of InfoLSTM is better than the best run of IMM enhanced model. The worst run of IMM enhanced model is better than the best run of IP model.

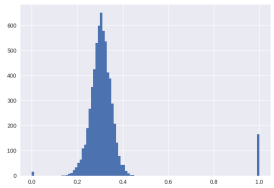
C. Visualization with t-SNE

Using t-SNE method [15], we map the 768-dimensional BERT embeddings to 2-dimensional space, and use shade of color to represent similarity. In the 2D space, location information represent the semantic relationship / text similarity to some extent. We choose two representative microblogs and draw the following four plot for them respectively:

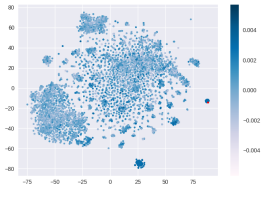
- In 2D space, all microblogs are drawn in the scatter plot, with the color representing the cosine similarity to the selected microblog (marked in red). Dark color means the two microblogs are very similar.
- The histogram of all cosine similarities to the selected microblog.
- We use color to represent the learned $\Delta_{cont.}(u_i, u_j)$ learned in IMM Enhanced model. Dark color means that reading u_j is making very positive impact on user decision to forward u_i. The coordinates of scatters are the same as the cosine one.
- The histogram of all learned $\Delta_{cont.}(u_i, u_j)$.



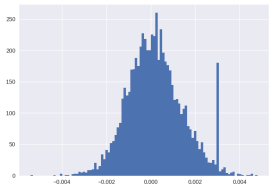
(a) Cosine Similarity w/ BERT Embeddings



(b) Histogram of Cosine Similarity



(c) $\Delta_{cont.}(u_i, u_j)$ Learned by IMM Enhanced Model



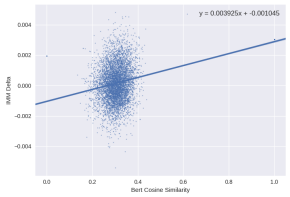
(d) Histogram of $\Delta_{cont.}(u_i, u_j)$

Fig. 7: Visualization of IMM Enhanced Model (id=1999)

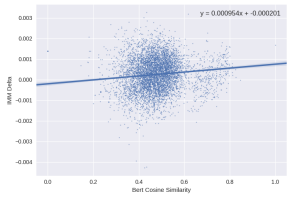
In Fig. 7 (a), we found that generally all microblogs form three big clusters (in the bottom-left, top, and middle). It shows that during the selected time period, some microblogs already form clusters automatically, and BERT pre-trained model are already able to capture text similarity. Also, from Fig. 7 (a) we found, there are two small cluster of microblogs which are very similar to the selected microblog (marked in red). From the histogram in Fig. 7 (b), there are several microblogs with similarity close to 1. For remaining microblogs, the similarity is below 0.4.

What is the relationship between text similarity and interaction during information diffusion? From Fig. 7 (c) we see that those having dark color in (a) still have dark color in (c). Meanwhile, the bar of cosine similarity of 1 in (b) is similar to the bar of interaction term of 0.003 in (d) with regard to quantity. To sum up, microblogs that are similar in content have positive impact in users' forwarding decision-making.

In Fig. 8, x-axis represent cosine similarity calculated with BERT embedding vectors, y-axis represent the interaction term $\Delta_{cont.}(u_i, u_j)$ learned by IMM enhanced model. We draw a scatter plot and run linear regression on these data. The result of linear regression is $y = 0.003925x - 0.001045$ and the correlation coefficient is 0.37. In general the two metrics are weakly but positively correlated. There may be other factors influencing content interaction, so that linear regression is not explaining the content interaction fully with text similarity, and it appears as the scatter plot contains much noise.



(a) Microblog ID 1999



(b) Microblog ID 3474

Fig. 8: Linear Regression between Text Similarity and Content Interaction

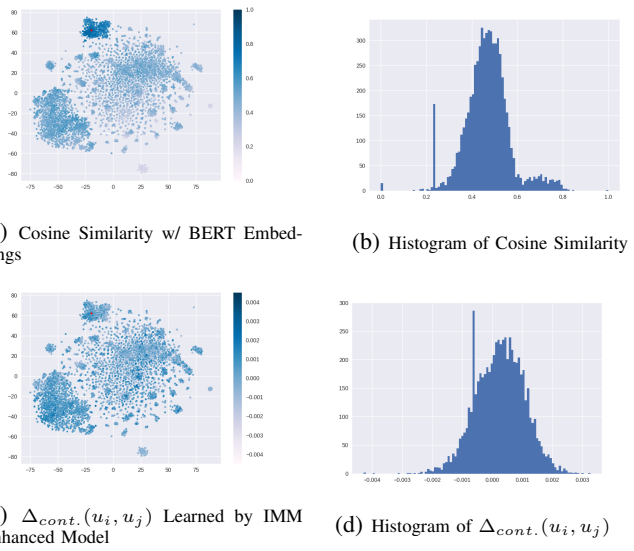
Observing the same plots of another selected microblog (id 3474) in Fig. 9, we have similar observations: (1) Those having high cosine similarity (in the top cluster in Fig. 9 (a)), have interaction terms of large positive value; (2) In other parts of the scatter plots, there are still microblogs with large interaction values, even though the text similarity is low. Similarly, we run linear regression and get the following result: $y = 0.000954x - 0.000201$. The correlation coefficient is 0.12, which is weaker but still positive.

To sum up, the relationship of text similarity and diffusion interaction can be concluded as follows:

- In the two linear regression attempts, the correlation coefficients are both positive. On average, we believe high text similarity leads to positive interaction in diffusion.
- However, this relation does not hold true in the opposite direction. Even though text similarity is low, two microblogs can still interact with each other significantly. As said in the introduction, the example of 'buying salt' + 'tsunami' and 'Wanzhou Meng' + 'Canadian Diplomat' are examples of two events with low text similarity influencing each other.

D. Merits and Demerits

We discuss the merits and demerits of the proposed InfoLSTM model and IMM enhanced model in this section. The merits include: (1) Performances of the two proposed



(a) Cosine Similarity w/ BERT Embeddings

(b) Histogram of Cosine Similarity

(c) $\Delta_{cont.}(u_i, u_j)$ Learned by IMM Enhanced Model

(d) Histogram of $\Delta_{cont.}(u_i, u_j)$

Fig. 9: Visualization of IMM Enhanced Model (id=3474)

models outperform those of baseline models. IMM model only improves the performance by 3% in F1 score, compared to IP model; however, the improvements made by InfoLSTM model and IMM Enhanced model over IMM model are significant. We believe IMM Enhanced model and InfoLSTM model better leverage the text information in training data and capture the content interactions in information diffusion. (2) By introducing novel BERT pre-trained model popular in natural language processing, IMM enhanced model and InfoLSTM model are more generalized. With original IMM model, predictions on a new, unseen microblog is infeasible since the corresponding vector M_i is not available; however, with BERT pre-trained model, we can get a 768-dimensional vector representing its semantics and feed it into the trained model for prediction. If BERT embeddings are set as untrainable during model training, the semantic meanings of BERT embeddings will remain consistent. (3) The interaction term $\Delta_{cont.}(u_i, u_j)$ learned by IMM Enhanced model has actual probabilistic meaning. A greater value means more positive impact to forwarding decision. These learned values can be further adopted in downstream tasks. For example, the current graphical evolutionary game framework for information diffusion is only applicable to single-information scenario. If we want to generalize it into multi-information scenario, content interaction may be introduced as input.

The demerits of these two models are summarized as follows: (1) InfoLSTM achieves the strongest performance; however, it is a still black-box model and the parameters learned are not interpretable. We still cannot explicitly understand how multiple pieces of information interact with each other. This is a common demerit of deep learning models. (2) In IMM model, each microblog corresponds to a vector, and the vector is learned from scartch (initialized with random number). Meanwhile, the number of times that each microblog appears in the dataset is limited and mostly very small; therefore some vector may not be sufficiently trained. We refer to this problem as 'the sparse problem'. We use stochastic gradient descent with mini-batch for optimization while training, and thus how these batches are generated will influence the final performance. This is manipulated by random seeds. This is the reason why the standard deviation of 5 runs of IMM model is comparatively large. With IMM enhanced model, even though the model has a better starting point with BERT pre-trained embeddings, 'the sparse problem' still exists and therefore the model's convergence is still affected by random seeds. Still, due to a better starting point, the worst run of IMM enhanced model is better than the best run of original IMM model. (3) Both proposed models strongly relies on hyper-parameters such as dropout rate, initial learning rate and hidden dimension. There's no empirical rule for selecting hyper-parameters. This poses a challenge for training models.

VI. Conclusion

In this paper, we focus on how multiple pieces of information interaction with each other in information diffusion process. In particular, we study this problem by modeling users' forwarding behavior when they have viewed a series of microblogs and is making decision on whether to forward the current microblog they're reading. Building upon existing models (IP model and IMM model), we introduce successful and novel achievements in deep learning – LSTM model and BERT pre-trained model, and design IMM enhanced model and InfoLSTM model for the task that we focus. IMM enhanced model is based on the math derivation of original IMM model. It also uses BERT pre-trained model to generate representation for each microblog's text. An extra linear layer is introduced to better fit the design of original IMM model. InfoLSTM generates a sequence of input with BERT pre-trained embeddings, and feeds it into a 2-layer LSTM network, which better simulates the memorizing behavior of users when browsing microblogs. InfoLSTM further uses logistic regression to do prediction.

On real-world Sina Weibo dataset, these two models achieves competitive prediction results, in which IMM Enhanced model achieves 52.24% of AUC and 55.81% of F1 score. InfoLSTM achieves 68.77% of AUC and 64.86% of F1 score. Compared with 26.96% AUC and 32.28% F1 score of original IMM model, the proposed two models prove to be more powerful. Comparing the precision-recall curves of all models, we also found that the proposed models outperform the baselines.

During our experiments, we found that the training process of InfoLSTM is more stable, while that of IMM model and IMM enhanced model is influenced by random seeds and may easily converge to local optima. Through our analysis, we suspect this is due to 'sparse' training data and insufficient training. Potential solutions include using other optimizer or initialize method, which we leave for future research.

An important merit of IMM enhanced model is that the learned parameters are interpretable. With the help of t-SNE method, we visualize the dataset and the learned parameters with scatter plots. We conduct case study with two selected

microblogs. We have two major conclusions: (1) On average, text similarity is positively correlated with context interaction, though the extent of correlation is different for each case; (2) High text similarity result in high content interaction, however this relationship does not hold true in the opposite direction – two microblogs with low text similarity may still have strong impact with each other in diffusion.

Our research is meaningful in various aspects. Firstly, as the prediction performance is improved, the trained models have more value in real-world applications such as commercial promotions and rumour control. Secondly, BERT pre-trained model is successfully applied in this forwarding behavior modeling task, which verifies the generalization ability of BERT model. Thirdly, the visualization and related analysis done in this paper help people better understand the relationship between text similarity and content interaction, quantitatively and qualitatively. Lastly, the results of IMM enhanced model can be further used in downstream tasks such as information diffusion modeling in graphical evolutionary game framework. The content interaction learned better suits the context in evolutionary game.

We hope our research can benefit future research, potentially in the following directions. (1) Though proves to be useful in previous work, personalized information is not our focus in this paper and thus is not used in the training process. Generating network embedding for users in a large network is also a popular research topic [16], and introducing user-related information will certainly improve model performance. (2) Attention is a novel mechanism widely used in deep learning recently [17]. It is inspired by how human can pinpoint key information when viewing pictures and reading text - people acquire such instinct through years of experience (i.e. large-scale training). When browsing microblogs, users have similar 'attention' – one may have deep impression on certain microblogs. As a result, attention can be introduced to LSTM layers in the model and provide interpretability. (3) One of our goal is to apply the learned model to unseen microblogs. This is important in some application scenarios where we need to train on existing data and do prediction on data collected in future. To better support these scenarios, dataset may be collected in a wider range of time and performance under different circumstances should be measured and discussed. (4) Graphical evolutionary game has its own characteristics in analyzing information diffusion. Currently its research is limited to single-information diffusion. Text similarity cannot fully describe content interaction, as discussed in our visualization and analysis. The IMM enhanced model can calculate the interaction term given the content of two microblogs, and may be used in the graphical evolutionary game framework for modeling simultaneous diffusion processs of multiple pieces of information in social network.

REFERENCES

[1] S. A. Myers and J. Leskovec, "Clash of the contagions: Cooperation and competition in information diffusion," in *2012 IEEE 12th international conference on data mining*, pp. 539–548, IEEE, 2012.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proceedings of the 13th international conference on World Wide Web*, pp. 491–501, ACM, 2004.

[4] C. Jiang, Y. Chen, and K. R. Liu, "Graphical evolutionary game for information diffusion over social networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 524–536, 2014.

[5] Y. Chen, C. Jiang, C.-Y. Wang, Y. Gao, and K. R. Liu, "Decision learning: Data analytic learning with strategic decision making," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 37–56, 2016.

[6] X. Zhang, Y. Su, S. Qu, S. Xie, B. Fang, and P. Yu, "Iad: interaction-aware diffusion framework in social networks," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

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

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[9] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[12] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, "Social influence locality for modeling retweeting behaviors," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[15] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[16] Y. Gu, Y. Sun, Y. Li, and Y. Yang, "Rare: Social rank regulated large-scale network embedding," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 359–368, International World Wide Web Conferences Steering Committee, 2018.

[17] 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*, pp. 5998–6008, 2017.

APPENDIX

A. Hyper-parameters

Table V and Table VI summarizes the hyper-parameters tuned with dev set and used in final model training.

TABLE V: Hyper-parameters used in InfoLSTM Model

Notation	Meaning	Value
d_b	Hidden Dimension of LSTM	200
p	Dropout Rate	0.1
α_0	Initial Learning Rate	1
b	Batch size	256

TABLE VI: Hyper-parameters used in IMM enhanced Model

Notation	Meaning	Value
d_h	Linear Layer Output Dimension	64
α_0	Initial Learning Rate	0.3
b	Batch size	256