Neural Conversation Recommendation with Online Interaction Modeling

Xingshan Zeng[1,2], **Jing Li**[3]*, **Lu Wang**[4], **Kam-Fai Wong**[1,2]
[1]The Chinese University of Hong Kong, Hong Kong, China
[2]MoE Key Laboratory of High Confidence Software Technologies, China
[3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[4]Khoury College of Computer Sciences, Northeastern University, Boston, United States
[1,2]{xszeng,kfwong}@se.cuhk.edu.hk
[3]jing-amelia.li@polyu.edu.hk, [4]luwang@ccs.neu.edu

Abstract

The prevalent use of social media leads to a vast amount of online conversations being produced on a daily basis. It presents a concrete challenge for individuals to better discover and engage in social media discussions. In this paper, we present a novel framework to automatically recommend conversations to users based on their prior conversation behaviors. Built on neural collaborative filtering, our model explores deep semantic features that measure how a user's preferences match an ongoing conversation's context. Furthermore, to identify salient characteristics from interleaving user interactions, our model incorporates graph-structured networks, where both replying relations and temporal features are encoded as conversation context. Experimental results on two large-scale datasets collected from Twitter and Reddit show that our model yields better performance than previous state-of-the-art models, which only utilize lexical features and ignore past user interactions in the conversations.

1 Introduction

Social media has profoundly revolutionized people's social interactions, as many individuals now turn to online platforms to voice opinions and exchange ideas. Meanwhile, the abundance of information brings the problem of information explosion — the huge volume of online discussions produced every day has far outpaced any individual's capability of digesting them. It is hence difficult for one to discover online discussions that are potentially of interest. To address this issue, we study the problem of online conversation recommendation, with the goal of identifying conversations that fit a user's preferences, hence likely to result in the user's future engagement.

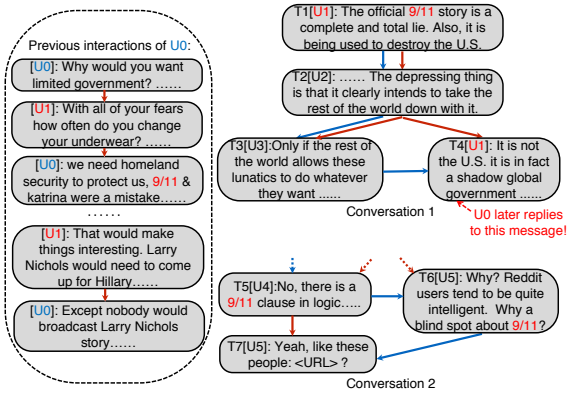


Figure 1: Two Reddit conversation snippets on the right. User U_0, whose historical interactions with another user U_1 shown on the left, only engages in Conversation 1 (which is initialized by U_1), but not Conversation 2 (U_1 does not participate in). Red arrows indicate in-reply-to relations, and blue arrows depict chronological orders.

In previous studies, it has been shown that effective online conversation recommendation has the potential to produce more positive online social interaction experience (Chen et al., 2011; Zeng et al., 2018). Prior work on this subject has focused on post-level recommendation (Yan et al., 2012; Chen et al., 2012), or conversation-level suggestion with handcrafted features (Chen et al., 2011) and word co-occurrence patterns (Zeng et al., 2018). Nevertheless, they ignore the useful information embedded in replying relations, where the conversation structure is formed via messages sent among users. In this work, we examine conversation context, and model the participants' interactions therein. This approach enables deep representation learning that reflects personal interests and conversation preferences, together signaling what conversations a user is likely to be involved in.

To illustrate how online interactions could indicate users' future conversation behavior, Figure 1

*This work was mainly conducted when Jing Li was affiliated with Tencent AI Lab, Shenzhen, China.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4632–4642, Hong Kong, China, November 3–7, 2019.

shows two conversation snippets on Reddit, both centering around the September 11 attack (9/11). As can be seen, user U_0, who had discussed the event according to the chat history, later engaged in Conversation 1 (C_1) instead of Conversation 2 (C_2). One explanation is that C_1 was initialized by user U_1, whose discussion topics overlap with U_0's, and more importantly, used to interact with U_0 in many prior discussions.

To model user preferences from their prior interactions, we propose to employ graph-structured neural networks to explicitly encode *who replies to whom at when* in the conversation history. In this way, temporal features of conversations are also exploited to capture messages' chronological orders (shown in Figure 1 with blue arrows). We then incorporate the interaction representations into a novel neural collaborative filtering framework (He et al., 2017), which further aligns user's preferences with the conversation context. Compared with existing methods that are based on handcrafted features (Chen et al., 2011) or Bayesian models (Zeng et al., 2018), our end-to-end trained neural model learns to automatically recommend conversations as well as to encode user interests embedded in their conversation interactions. To the best of our knowledge, this is the first work to explore neural conversation recommendation with online interactions explicitly encoded for user preference modeling.

To evaluate our model, we conduct extensive experiments on two large-scale datasets with online conversations from Twitter and Reddit[1]. Experimental results show that our method significantly outperforms state-of-the-art models that do not capture user interactions. For example, our model obtains an MAP (Mean Average Precision) of 0.625 on Twitter, compared with 0.591 by Zeng et al. (2018). We further find that our model still exhibits superior performance when the sparsity levels of user history and conversation context are varied, demonstrating our model's potential ability to handle sparse conversation records. Additional experiments on an ablation study confirms the effectiveness of different components in our framework. A case study further reveals important interaction features captured by our model, which indicate their conversation entries and hence explain our model's advanced performance. Finally, we investigate the challenging task of first time replies prediction, where our model again produces significantly better results than existing popular recommendation models.

[1]The datasets and codes are available at: https://github.com/zxshamson/neural-conv-rec

2 Related Work

Our work is in line with conversation behavior analysis, where studies explore user interactions in ongoing conversations (Ritter et al., 2010) and how they signal the conversations' future trajectory, such as continued activity (Backstrom et al., 2013; Jiao et al., 2018; Zeng et al., 2019) and the risk of going awry (Zhang et al., 2018). Different from these proposals which do not model personal interests, we study conversation recommendation for a specific user, where we measure how a user's preferences match a conversation's context.

This work is also related to user response prediction (Artzi et al., 2012; Zhang et al., 2015) and post recommendation (Duan et al., 2010; Chen et al., 2012; Yan et al., 2012; Hong et al., 2013). While most of these studies focus on post modeling, we examine conversation context to predict user engagements, which goes beyond the post-level prediction task. Other prior work examining conversation-level recommendation relies on either manual features (Chen et al., 2011) or shallow word occurrence patterns (Zeng et al., 2018), largely ignoring the useful features from historical user interactions. On the contrary, we utilize online user interactions in the conversation history, to allow the inclusion of richer information of modeling personal interests. In addition, our neural network-based model enables automatic learning for a deeper representation of user interests, whereas existing methods require significant manual efforts for model customization (Chen et al., 2011; Zeng et al., 2018).

Furthermore, our user interaction module is inspired by prior work on conversation structure modeling. Compared with popular sequential conversation models that focus on messages' temporal features (Cheng et al., 2017; Jiao et al., 2018; Zeng et al., 2019), our module explicitly encodes the replying relationships to exploit the user conversation structure (Miura et al., 2018; Zayats and Ostendorf, 2018). It is shown that such structure indicates salient messages and can benefit various compelling applications, e.g., conversation summarization (Chang et al., 2013; Li et al., 2015) and discussion topic extraction (Li et al., 2016, 2018).

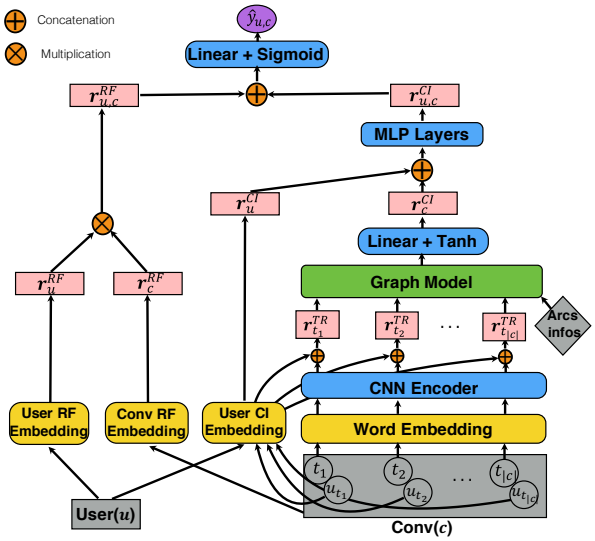


Figure 2: The architecture of our neural conversation recommendation framework, which models replying preferences and conversation interactions to predict a user's future engagement in given conversations. RF: replying factors modeling. CI: conversation interaction modeling.

However, its effect on conversation recommendation has not been explored yet, and our work aims to fill the gap.

3 Neural Conversation Recommendation

This section describes our neural recommendation model with interaction modeling. Figure 2 shows the overall architecture of our framework based on neural collaborative filtering (NCF) (He et al., 2017). Section 3.1 will present an overview showing how our model works, where both users' replying history and conversations' interaction structure will be encoded for recommendation. Their modeling details will be given in Section 3.2 and 3.3 in turn. At last, Section 3.4 shows the overall model training process.

3.1 Model Overview

Here we first describe the input and output. For training, our model is fed with a conversation dataset $\mathcal{C}$. Each conversation $c \in \mathcal{C}$ is formed with a sequence of turns $\langle t_1, t_2, ..., t_{|c|} \rangle$, where $|c|$ denotes the number of turns. Each turn t is in form of a word sequence $\langle w_1, w_2, ..., w_{|t|} \rangle$ with $|t|$ being t's word number. Its author is represented by user id u_t. We also record each turn's parent turn in replying relations (i.e. which turn it replies to) and chronological order (i.e. which turn posted before it), so that the interaction patterns within a conversation can be captured. We will talk more about it in Section 3.3.

For recommendation, our model is taken a user u and a conversation c as input, and then predict how likely u will engage in c, conditioned on u's previous behavior and c's context history.

Our goal is to predict $\hat{y}_{u,c} \in [0, 1]$, which measures how likely user u will engage in conversation c. Here to estimate $\hat{y}_{u,c}$, two types of information are encoded: replying history of users and interaction structure of conversations. The former is captured from what conversations a user previously replied to, where we learn $\boldsymbol{r}^{RF}_{u,c}$ to encode u's replying preference on c. However, such learned representation captures user replying preferences without diving into turn-level features and interaction structure in conversations. So we utilize the latter to explore how users interact with each other in conversation context, which encodes words in turns and turn interactions to produce conversation interaction representation $\boldsymbol{r}^{CI}_{u,c}$, reflecting a denser preference of a user. In Section 3.2, we will present how to learn $\boldsymbol{r}^{RF}_{u,c}$, and in Section 3.3 we learn about $\boldsymbol{r}^{CI}_{u,c}$.

Coupling the $\boldsymbol{r}^{RF}_{u,c}$ and $\boldsymbol{r}^{CI}_{u,c}$, we predict $\hat{y}_{u,c}$ via the formula below:

$$\hat{y}_{u,c} = \sigma(\boldsymbol{h}_O^T[\boldsymbol{r}^{RF}_{u,c}; \boldsymbol{r}^{CI}_{u,c}]) \quad (1)$$

where $\sigma(\cdot)$ denotes sigmoid activation, $[;]$ indicates concatenation operation, and $\boldsymbol{h}_O$ is a learnable parameter. In recommendation for user u, we rank the conversations with $\hat{y}_{u,c}$ and the top N results will serve as our final output.

3.2 Replying Factors Modeling

As mentioned in Section 3.1, we first model users' replying preferences with what conversations they entered before. We follow the practice in He et al. (2017) to use two embedding layers, $I^{RF}_U(\cdot)$ and $I^{RF}_C(\cdot)$, to capture the latent factors for users and conversations that result in user's previous replying history. For user u, we can obtain its user embedding $\boldsymbol{r}^{RF}_u$ by looking up u in $I^{RF}_U(\cdot)$. A conversation embedding $\boldsymbol{r}^{RF}_c$ can be similarly obtained from $I^{RF}_C(\cdot)$. Then we measure user u's replying preference over conversation c with the similarity between $\boldsymbol{r}^{RF}_u$ and $\boldsymbol{r}^{RF}_c$:

$$\boldsymbol{r}^{RF}_{u,c} = \boldsymbol{r}^{RF}_u \odot \boldsymbol{r}^{RF}_c \quad (2)$$

where $\odot$ denotes element-wise product. As can be seen, $\boldsymbol{r}^{RF}_{u,c}$ is able to encode what conversations

a user engages in and analyze the factors of why it happen, simply with general replying history. More features will be explored via conversation interaction modeling presented in Section 3.3.

3.3 Conversation Interaction Modeling

We first explore users' prior interaction behavior in conversations. A user embedding layer $I_U^{CI}(\cdot)$ is hence employed, where the embedding $\boldsymbol{r}_u^{CI}$ for user u reflects u's interaction patterns, such as what they used to say and whom they usually interacted with. For the conversation modeling, we adopt graph-structured networks to model the interaction structure therein and yield a representation $\boldsymbol{r}_c^{CI}$ for conversation c. The effects of user- and conversation-specific interaction features are combined with Multilayer Perceptron (MLP):

$$\boldsymbol{r}_{u,c}^{CI} = \alpha(W_M^T(...\alpha(W_1^T[\boldsymbol{r}_u^{CI}; \boldsymbol{r}_c^{CI}] + b_1)...) + b_M) \quad (3)$$

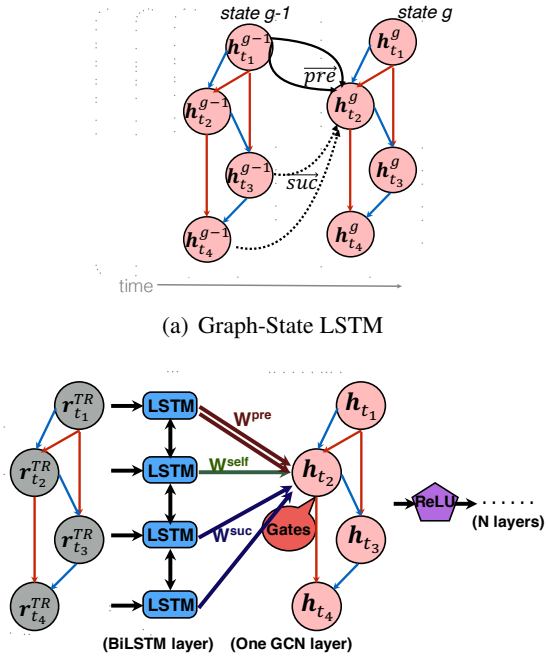
where $\alpha(\cdot)$ is ReLU-activated function (Rectified Linear Unit) and M is the number of layers in MLP. In the following, we will introduce how we obtain $\boldsymbol{r}_c^{CI}$ via modeling of intra-turn features and inter-turn interactions.

Turn-level Modeling. Here we describe how we model turn-level representations, which combine what content it conveys and who its author is.

Content representation is to reflect how words appear therein, where we employ a Convolutional Neural Network (CNN) (Kim, 2014) encoder to model a turn's word sequence. Specifically, given a turn t in conversation c, we first map each word in t into a word embedding layer (initialized with pre-trained word vectors) to explore deep word semantics. And then, to capture how a word appears in local context with its neighbors, a CNN encoder is exploited to generate the turn-level content representation $\boldsymbol{z}_t$.

Next, we concatenate $\boldsymbol{z}_t$, conveying content features, and $\boldsymbol{r}_{u_t}^{CI}$, embedded with the interaction patterns of t's author u_t, to produce a turn representation $\boldsymbol{r}_t^{TR}$. It couples turn t's word occurrence patterns and its author's history interactions with other conversation turns. Afterwards, $\boldsymbol{r}_t^{TR}$ is delivered to model t's interaction with the other turns in c. We will describe how it is processed next.

Turn Interaction Modeling. To encode conversation interaction structure, we first organize the turns in a conversation c as a reply tree to formulate who replies to whom. Each node therein represents a turn and the edges reflect replying relations (directed from turns to replies such as the red arrows in Figure 1). Moreover, to exploit temporal information, we add another kind of edges to indicate chronological order (such as the blue arrows in Figure 1). In doing so, a reply tree is extended to a directed graph (such as the one in Figure 1), with both replying and temporal interactions encoded and therefore named as an interaction graph. For each turn t on the graph, we distinguish its neighbors into predecessors, denoted by $E^p(t)$, and successors, $E^s(t)$.



(a) Graph-State LSTM

(b) Graph Convolutional Networks

Figure 3: Two variants of our proposed graph-structured networks.

Then, we employ graph-structured networks to model the interaction structure. There are two modeling methods discussed here: **Graph-State LSTM** (Long Short-Term Memory) (henceforth GLSTM) (Beck et al., 2018; Song et al., 2018) and **Graph Convolutional Networks** (henceforth GCN) (Kipf and Welling, 2017; Marcheggiani and Titov, 2017), whose empirical effectiveness will be compared in Section 5.1. Here we present their architecture in Figure 3 and describe how they model conversation interactions below.

Graph-State LSTM. We start with GLSTM and show its architecture in Figure 3(a). It is an extension of LSTM from sequence to graph structure, where a turn's hidden states are updated conditioned on both the turn-level representation $\boldsymbol{r}_t^{TR}$

and the states of all its neighbors on the graph. The update strategy is the same as standard LSTM (Hochreiter and Schmidhuber, 1997), except for the following formula, which can be used in the update of input gate, output gate, forget gate, and content recorder:

$$\boldsymbol{g}_t^{GLSTM} = \sigma(W^p \boldsymbol{x}_t^p + W^s \boldsymbol{x}_t^s + U^p \boldsymbol{h}_t^p + U^s \boldsymbol{h}_t^s + b) \quad (4)$$

The first two terms explore the turn-level representations ($\boldsymbol{r}_t^{TR}$) from the neighbors. The third and forth terms capture turn interactions on the graph. b denotes the bias. The superscripts p and s indicate the neighbor being a predecessor or successor. $\boldsymbol{x}_t^p$ takes the sum of predecessor k's turn representations $\boldsymbol{r}_k^{TR}$ and so does $\boldsymbol{x}_t^s$ for successors. $\boldsymbol{h}_t^*$ means the neighbors' hidden states in their last updates. W^* and U^* are learnable parameter weights and $\sigma(\cdot)$ means sigmoid activation. Moreover, in GLSTM, we define the state number g to reflect the maximum order of GLSTM state transitions, where the larger g indicates longer turn dependency on graph paths encoded. Here due to the space limitation, we leave out the details of GLSTM and refer the readers to Song et al. (2018).

Afterwards, to produce conversation representation $\boldsymbol{r}_c^{CI}$ with turn interactions, we combine all turns' hidden states with average pooling and map them into the same dimension (with Tanh activation) as $\boldsymbol{r}_u^{CI}$ to measure user and conversation similarity.

Graph Convolutional Networks. Figure 3(b) shows the architecture of GCN, which can be considered as CNN on graph. Here following previous practice (Marcheggiani and Titov, 2017), before using GCN to model turn interactions, we first feed the turn representations $\boldsymbol{r}_t^{TR}$ into a sequential Bidirectional LSTM (BiLSTM) layer to capture the chronological turn interactions. Then, we take the t-th hidden states of BiLSTM $\boldsymbol{h}_t^{LSTM}$ to further capture turn t's interaction with its neighbors on interaction graph. The formula describing this process is given as:

$$\begin{aligned}\boldsymbol{h}_t^{GCN} = &\sum_{i \in E^p(t)} \omega_{i,t}(W^{pre}\boldsymbol{h}_i^{LSTM} + b^{pre}) + \\ &\sum_{j \in E^s(t)} \omega_{j,t}(W^{suc}\boldsymbol{h}_j^{LSTM} + b^{suc}) + \\ &\omega_{t,t}(W^{self}\boldsymbol{h}_t^{LSTM} + b^{self})\end{aligned} \quad (5)$$

Here following Marcheggiani and Titov (2017), we use different sets of parameters to fit varying types of interactions: self interactions (self), interaction from predecessors to successors (pre), and that from the other way around (suc). $\omega_{i,j}$ is a scalar gate controlling weights defined below:

$$\omega_{i,j} = \sigma(\boldsymbol{h}_i^{LSTM} \cdot V^{dir(i,j)} + d^{dir(i,j)}) \quad (6)$$

It is to identify the neighbors affecting more to t than others. $dir(i,j)$ indicates the type of i-j direction (pre, suc, or self).

Furthermore, to allow deep interactions to be learned, we can stack multiple GCN layers to form a multi-layer GCNs, where we apply a ReLU activated function between two layers. After that, we take the similar operations as for GLSTM yield conversation representation $\boldsymbol{r}_c^{CI}$ as c's conversation interaction representation.

3.4 Model Training

Here we describe how we formulate our learning objective and train our model. As mentioned above, our goal is to predict a score $\hat{y}_{u,c} \in [0,1]$ indicating how likely user u will reply to conversation c. In training, we adopt binary cross-entropy as our learning loss with penalty given to negative feedback (u does not engage in c). It is because negative feedback may happen for many unpredictable reasons, such as users being too busy to go online. Thus for conversation recommendation, we rely more on the positive feedback and design the *weighted binary cross-entropy loss* below:

$$\mathcal{L} = -\sum_{(u,c)\in\mathcal{T}} \left[\lambda \cdot y_{u,c}\log(\hat{y}_{u,c}) + (1-y_{u,c})\log(1-\hat{y}_{u,c})\right] \quad (7)$$

where $\mathcal{T}$ is a set of training instances. $y_{u,c}$ is a binary label indicating whether u replied to c, and $\hat{y}_{u,c}$ is our predicted score. λ ($\lambda > 1$) is a predefined parameter to trade off the weights of positive and negative instances.

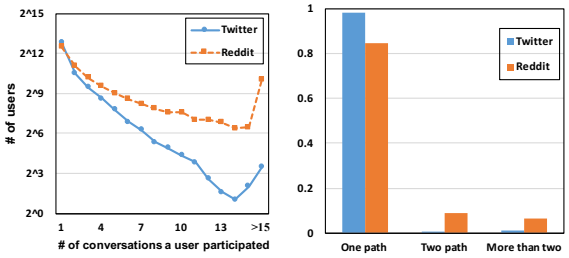
In model training, the negative sampling strategy is adopted (He et al., 2017), whose sampling ratio (the number of negative samples for each positive instance) is set to 5. Also, we pre-train the embedding layers for both the replying factors and conversation interaction modeling with the parameters from He et al. (2017). We will discuss the effects of pre-training in Section 5.3.

4 Experimental Setup

Data Collection and Preprocessing. In our experiments, we use datasets from two different platforms: the first one is released by Zeng et al. (2018) containing **Twitter** conversations formed by tweets from the TREC 2011 microblog track

	Twitter	Reddit
# of users	10,122	13,134
# of conversations	7,500	29,477
# of turns	38,999	109,774
Avg. # of conversations per user	1.7	5.9
Avg. # of turns per conversation	5.2	3.7
Avg. # of users per conversation	2.3	2.6

Table 1: Dataset statistics.



(a) User conversation dist. (b) Ratio of diff. structure

Figure 4: Distribution of number of conversations per user (left) and proportion of different conversation structures (right) for both datasets.

data[2] covering a diverse set of topics; the other is from Zeng et al. (2019), which is comprised of discussion threads about political issues on **Reddit**, a popular discussion website.

The tweets in Twitter dataset were mainly posted from Jan 23 to Feb 8, 2011, and discussion threads in Reddit dataset were posted from Jan to Dec, 2008. To discover the whole conversations, we retrieved all messages with replying relations (indicated by "*parent_id*" property in Reddit corpus, for example), and recorded their authors and parent messages. Finally, conversations with only one message were removed.

We applied the Glove tweet preprocessing toolkit (Pennington et al., 2014)[3] on the Twitter dataset. As for the Reddit dataset, we performed tokenization using open source natural language toolkit (NLTK) (Loper and Bird, 2002), with links replaced to a generic tag "URL" and all number tokens removed. We maintained a vocabulary with all the rest characters appearing in the corpus for both datasets, including punctuation and emoticons.

Data Statistics and Analysis. The statistics of two datasets are shown in Table 1, with more information in Figure 4. We can observe that Reddit dataset contains more conversations, with a higher average number of conversations per user. On the other hand, Twitter conversations are longer, with fewer participants. Figure 4(a) shows that most users participate in very few conversations in both datasets, indicating a potential sparsity problem. In terms of conversation structure (Figure 4(b)), most conversations only contain one path where the replying relations precisely follow the chronological order; whereas the Reddit dataset contains more tree-structured conversations with rich and complex interactions.

To further illustrate the effect of a conversation's structure on its future development, we calculate the likelihoods of (1) new users joining the discussion, and (2) current participants continuing the conversation, grouped by different conversation structures (Table 2). In general (especially on Reddit), conversations with only one path tend to continue within the current participants, and keep newcomers out, whereas conversations with complex structures are more likely to attract new users.

	Twitter			Reddit		
Paths amount	1	2	> 2	1	2	> 2
New comers rate	0.18	0.20	0.16	0.30	0.45	0.50
Continue rate	0.70	0.64	0.74	0.51	0.31	0.38

Table 2: Different structures and the future trend of conversations.

Model Setting. We follow the experimental settings employed in previous work (Zeng et al., 2018). For each conversation, we take first 75% of the context as observation for training purpose. The rest is equally divided into a testing set and a development set. For negative instances, we also split the unobserved user-conversation pairs into three parts: 80%, 10%, and 10% for training, testing, and development, respectively. Furthermore, due to the large amount of conversations in Reddit dataset, we only sample 100 negative instances uniformly from them for testing and development.

For parameters setups, we initialize the word embedding layer with 200-dimensional Glove embedding (Pennington et al., 2014), where the Twitter version is used for our Twitter dataset, and the Common Crawl version is applied on the Reddit dataset[4]. Factor dimension for the RF part is set to 20, while for the CI part it is 100. For the CNN

[2] https://trec.nist.gov/data/tweets/

[3] https://nlp.stanford.edu/projects/glove/preprocess-twitter.rb

[4] https://nlp.stanford.edu/projects/glove/

encoders, we use filter windows of 2, 3, and 4, each with 100 feature maps. For the size of hidden states of our graph models, we set 200 (100 for each direction for BiLSTM). The number of MLP layers is 3. During training, the batch size is set to 512 and Adam optimizer (Kingma and Ba, 2014) is adopted with an initial learning rate of 0.01. We set the trade off weight in learning loss $\lambda = 100$.

Evaluation and Comparisons. Following Zeng et al. (2018)'s work, we adopt mean average precision (MAP), precision at 1 (P@1), and normalized Discounted Cumulative Gain at 5 (nDCG@5) for evaluation (we also try other metrics including P@5 and nDCG@10, and find similar trends). The metrics are first computed for users in the datasets, then averaged over all users.

For comparison, we first search for the best model among different interaction modeling (Section 5.1). We then consider two baselines: 1) ranking conversations randomly (RANDOM), 2) conversations with more participants ranked higher (POPULARITY). Previous work compared includes[5]:

• RSVM: Ranks conversations for each user with features described in Duan et al. (2010) by ranking SVM (Joachims, 2002).

• NCF: The neural CF model (He et al., 2017), not utilizing any context information.

• CONVMF: A CNN-based model for recommendation with reviews (Kim et al., 2016), where we adapt to use a hierarchical two-layer CNN to model words in turns and turn sequences.

• CR_JTD: The state-of-the-art method for our task (Zeng et al., 2018), with a Bayesian model jointly modeling topics and discourse.

5 Experimental Results

In this section, we first evaluate the effectiveness of varying modules for conversation interaction modeling in Section 5.1. Then our model with the best module is further compared with the baselines and previous recommendation systems in Section 5.2. There we also discuss the model performance given varying conversation context length and user interaction sparsity. We further discuss our model with an ablation study and a case study in Section 5.3. Finally in Section 5.4, we analyze the results of first time replies prediction.

[5]We did not compare with Chen et al. (2011) since that method mainly requires information about social relationship, which is unavailable in our datasets.

Models	Train Time	MAP	
		Twitter	Reddit
BiLSTM	0.94	0.617	0.498
GLSTM	1.25	0.617	0.528
GCN (W/O BiLSTM)	1.03	0.619	0.530
GCN (With BiLSTM)	1.00	**0.620**	**0.533**

Table 3: Results of our model variants on development set. The best MAP results are in **bold**. "Train Time": training time per epoch divided by that of model GCN (With BiLSTM).

5.1 Interaction Modeling Comparison

We first compare the effects of varying interaction modeling methods (see Section 3.3) on conversation recommendation. Table 3 displays their results on development set. In comparison, we consider BiLSTM over turn sequence (only chronological order encoded and henceforth BiLSTM), GLSTM (state number $g = 6$), GCN (layer number set to 3) without BiLSTM-encoded temporal representations (henceforth GCN (W/O BiLSTM)), and the full GCN described in Section 3.3 (henceforth GCN (With BiLSTM) and layer number set to 1). The above hyper-parameters are tuned based on the training loss.

From the results, we find that BiLSTM exhibits the worst results for not encoding replying relations. Its difference from others are larger on Reddit attributed to the rich replying structure therein (as shown in Figure 4(b)). The best performance is achieved for GCN (With BiLSTM), with relatively less training time. This shows the effectiveness and efficiency to explore the order of turns with BiLSTM and the user interactions with GCN. In the later analysis, we will only discuss our model that exploits GCN (With BiLSTM) for interaction modeling.

5.2 Comparisons with Previous Work

Main Results. Table 4 shows the conversation recommendation results with baselines and state of the arts. Our model exhibits the best results on both datasets, significantly outperforming all the comparison models. It indicates the usefulness to encode user interactions for conversation recommendation. Particularly, CONVMF is able to encode turns' temporal orders yet ignores how they reply with each other in conversation history. It is outperformed by our model, showing the benefit to capture users' replying patterns for predicting what conversations will draw their engagement.

Models	Twitter			Reddit		
	MAP	P@1	nDCG	MAP	P@1	nDCG
Baselines						
RANDOM	0.006	0.001	0.002	0.040	0.010	0.022
POPULARITY	0.023	0.005	0.010	0.082	0.033	0.063
Comparisons						
RSVM	0.554	0.575	0.559	0.453	0.457	0.466
NCF	0.573	0.593	0.576	0.412	0.544	0.461
CONVMF	0.579	0.596	0.583	0.485	0.532	0.520
CR_JTD	0.591	0.591	0.600	0.453	0.559	0.485
OURS	**0.625**	**0.632**	**0.626**	**0.538**	**0.674**	**0.590**

Table 4: Main results on conversation recommendation. "nDCG" stands for "nDCG@5". The best result for each column is in **bold**. Our model significantly outperforms all the comparisons ($p < 0.01$, paired t-test).

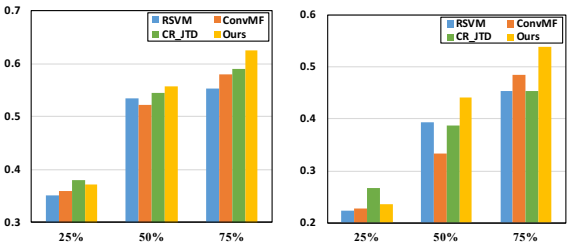
We also observe that both baseline models work poorly. It is because conversation recommendation is challenging, not possible to be well tackled with simple ranking strategies.

In addition, we notice that CR_JTD outperforms CONVMF on Twitter, with opposite observation made on Reddit. It is possibly because Twitter exhibits more informal language style. Thus CR_JTD, taken bag-of-words input, can better fit the data than CONVMF, taking word orders into account. Nevertheless, our model outperforms them both, showing that prior interactions among users can better signal their future reply behavior compared with the words they said.

The final observation is that, all comparing methods (except for the naive baselines) perform better on Twitter than Reddit. One reason is that the Twitter dataset is smaller and contains fewer users and conversations. Another possible reason might be that the topics in the Reddit dataset are mostly about politics, while the Twitter conversations are of diverse topics, which makes the model easier to distinguish user interests.

Training with Varying Conversation History. The main results are reported given the first 75% turns as conversation history. Here we investigate how the length of conversation history affects reply preference prediction. The models are hence trained with the first 25%, 50%, and 75% turns as conversation history and their MAP scores are shown in Figure 5.

As can be seen, all models exhibit better results when trained with longer history. This shows that users' future conversation preference can be better predicted with richer history data. We also observe



(a) Twitter (b) Reddit

Figure 5: MAP scores (in Y-axis) of models trained with the first 25%, 50%, and 75% turns as history.

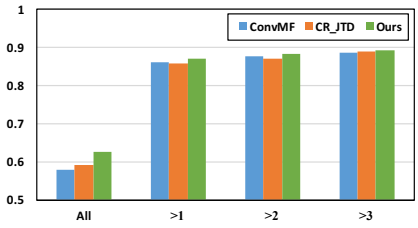


Figure 6: MAP scores (in Y-axis) for users with varying degrees of interaction history. X-axis indicates the number of conversations users previously engage in. *"All"* means recommendation for all users.

that our model obtains the best MAP on both the 50% and 75% setting, while for 25%, it is outperformed by CR_JTD. It might be ascribed to the sparse user interactions exhibited in 25% context, where there are only 1 or 2 turns on average according to Table 1.

Results for Varying User Interaction Sparsity. Figure 4(a) has shown that most users only engage in very few conversations. This results in severe data sparsity in user interaction history, especially on Twitter. We are hence interested in how models perform on varying degree of sparsity.

Figure 6 shows the MAP scores in recommendation to users engaged in varying number of conversations before on Twitter. As can be seen, user interaction sparsity can largely affect recommendation performance, where all models perform poorly for users exhibiting less than one conversation entry. We also observe that our model performs consistently better in varying degrees of sparsity. It may because our model is able to learn rich interactions from conversation context, which helps alleviate the sparsity in user history.

5.3 Further Discussion

Here we further discuss what our model learns leading to its superiority.

Ablation Study. We start with an ablation study to discuss the relative contributions of our different components. The MAP scores of their ablations are compared in Table 5. Our full model performs the best, showing that all components are useful. It is also seen that RF modeling contribute the most, meanwhile better contributions can be made with its parameters pre-trained. It indicates the crucial role RF modeling plays for neural conversation recommendation.

Models	Twitter	Reddit
W/O RF modeling (Sec. 3.2)	0.509	0.239
W/O pre-training (Sec. 3.4)	0.593	0.525
W/O MLP layers (Eq. 3)	0.600	0.537
W/O gate control weights (Eq. 6)	0.608	0.530
Our full model	**0.625**	**0.538**

Table 5: MAP scores obtained by our ablations. The best MAP results are highlighted in **bold**.

Case Study. To further understand how our model predicts users' conversation preferences, we take the example in Figure 1 and analyze what our model learns for it. Recall that user U_0 used to discuss a lot on 9/11 with U_1, who starts C_1. U_0 later engages in C_1 instead of C_2, though it also concerns 9/11.

	U_1	U_2	U_4	U_5
$\boldsymbol{r}_u^{RF}$	**0.761**	0.731	0.681	0.659
$\boldsymbol{r}_u^{CI}$	**0.763**	0.453	0.287	0.737

Table 6: Cosine similarity between U_0's user embeddings in RF and CI modeling with others'.

Table 6 shows the similarity of user factors learned by our RF (replying factor modeling in Section 3.2) and CI (conversation interaction modeling in Section 3.3) modules. As can be seen, both modules learn that U_0 and U_1 are similar (probably referred from their frequent interactions). Their joint effects result in our successful prediction of U_0 to engage in C_1 rather than C_2. For the same reason, our model can further predict that U_0 is more likely to reply to T_4 (in C_1), which is posted by U_1, based on the similarity between user factors and turn representations.

5.4 First Time Replies Prediction

In some scenarios, users may be more interested in seeing new conversations, which they haven't seen before but potentially match their preferences. We hence examine model performance to predict only first time replies and show their MAP scores in Table 7. It is observed that all models perform poorly, which implies that recommending unseen conversations to users is extremely difficult. This finding is consistent with Zeng et al. (2018). However, our model still outperforms others by a large margin. It again demonstrates the effectiveness of modeling user interactions for recommendation.

Models	Twitter	Reddit
RSVM	0.002	0.049
NCF	0.033	0.038
ConvMF	0.049	0.210
CR_JTD	0.090	0.075
Ours	**0.160**	**0.212**

Table 7: MAP scores for first time replies prediction.

6 Conclusion

We study neural conversation recommendation with graph-structured networks to encode user interactions. Experimental results on Twitter and Reddit show our model significantly outperforms the state of the arts. We also observe that competitive results can still be obtained on varying conversation history length and user interaction sparsity. Further discussions analyze the contributions of different components of our model and the useful features we learn leading to our superiority. At last, we study a challenging task of first time replies prediction, where our model still exhibits its effectiveness.

Acknowledgements

This work is partially supported by the following HK grants: RGC-GRF (14232816, 14209416, 14204118, 3133237), NSFC (61877020) & ITF (ITS/335/18). Lu Wang is supported in part by National Science Foundation through Grants IIS-1566382 and IIS-1813341. We thank the three anonymous reviewers for the insightful suggestions on various aspects of this work.

References

Yoav Artzi, Patrick Pantel, and Michael Gamon. 2012. Predicting responses to microblog posts. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 602–606. Association for Computational Linguistics.

Lars Backstrom, Jon M. Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. 2013. Characterizing and curating conversation threads: expansion, focus, volume, re-entry. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 13–22.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 273–283.

Yi Chang, Xuanhui Wang, Qiaozhu Mei, and Yan Liu. 2013. Towards twitter context summarization with user influence models. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 527–536.

Jilin Chen, Rowan Nairn, and Ed Huai-hsin Chi. 2011. Speak little and well: recommending conversations in online social streams. In *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 217–226.

Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. 2012. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR Conference on Research and development in information retrieval*, pages 661–670. ACM.

Justin Cheng, Michael Bernstein, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 1217–1230. ACM.

Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. 2010. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 173–182.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Liangjie Hong, Aziz S Doumith, and Brian D Davison. 2013. Co-factorization machines: modeling user interests and predicting individual decisions in Twitter. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pages 557–566. ACM.

Yunhao Jiao, Cheng Li, Fei Wu, and Qiaozhu Mei. 2018. Find the conversation killers: A predictive study of thread-ending posts. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1145–1154. International World Wide Web Conferences Steering Committee.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142.

Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 233–240.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Jing Li, Wei Gao, Zhongyu Wei, Baolin Peng, and Kam-Fai Wong. 2015. Using content-level structures for summarizing microblog repost trees. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2168–2178.

Jing Li, Ming Liao, Wei Gao, Yulan He, and Kam-Fai Wong. 2016. Topic extraction from microblog posts using conversation structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Jing Li, Yan Song, Zhongyu Wei, and Kam-Fai Wong. 2018. A joint model of conversational discourse and latent topics on microblogs. *Computational Linguistics*, 44(4).

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1506–1515.

Yasuhide Miura, Ryuji Kano, Motoki Taniguchi, Tomoki Taniguchi, Shotaro Misawa, and Tomoko Ohkuma. 2018. Integrating tree structures and graph structures with neural networks to classify discussion discourse acts. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3806–3818.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 172–180.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1616–1626.

Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 516–525. Association for Computational Linguistics.

Victoria Zayats and Mari Ostendorf. 2018. Conversation modeling on reddit using a graph-structured LSTM. *TACL*, 6:121–132.

Xingshan Zeng, Jing Li, Lu Wang, Nicholas Beauchamp, Sarah Shugars, and Kam-Fai Wong. 2018. Microblog conversation recommendation via joint modeling of topics and discourse. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)*, pages 375–385.

Xingshan Zeng, Jing Li, Lu Wang, and Kam-Fai Wong. 2019. Joint effects of context and user history for predicting online conversation re-entries. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2809–2818.

Justine Zhang, Jonathan P Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. 2018. Conversations gone awry: Detecting early signs of conversational failure. *arXiv preprint arXiv:1805.05345*.

Qi Zhang, Yeyun Gong, Ya Guo, and Xuanjing Huang. 2015. Retweet behavior prediction using hierarchical Dirichlet process. In *AAAI*, pages 403–409.