Modeling Global and Local Interactions for Online Conversation Recommendation

XINGSHAN ZENG, The Chinese University of Hong Kong, China

JING LI, The Hong Kong Polytechnic University, China

LINGZHI WANG and KAM-FAI WONG, The Chinese University of Hong Kong, China

The popularity of social media platforms results in a huge volume of online conversations produced every day. To help users better engage in online conversations, this article presents a novel framework to automatically recommend conversations to users based on what they said and how they behaved in their chatting histories. While prior work mostly focuses on post-level recommendation, we aim to explore conversation context and model the interaction patterns therein. Furthermore, to characterize personal interests from interleaving user interactions, we learn (1) *global interactions*, represented by topic and discourse word clusters to reflect users' content and pragmatic preferences, and (2) *local interactions*, encoding replying relations and chronological order of conversation turns to characterize users' prior behavior. Built on collaborative filtering, our model captures global interactions via discovering word distributions to represent users' topical interests and discourse behaviors, while local interactions are explored with graph-structured networks exploiting both reply structure and temporal features. Extensive experiments on three datasets from Twitter and Reddit show that our model coupling global and local interactions significantly outperforms the state-of-the-art model. Further analyses show that our model is able to capture meaningful features from global and local interactions, which results in its superior performance in conversation recommendation.

CCS Concepts: • **Information systems** → **Recommender systems**; **Personalization**; **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Conversation recommendation, graph neural networks, probabilistic graphical models

ACM Reference format:

Xingshan Zeng, Jing Li, Lingzhi Wang, and Kam-Fai Wong. 2021. Modeling Global and Local Interactions for Online Conversation Recommendation. *ACM Trans. Inf. Syst.* 40, 3, Article 49 (November 2021), 33 pages. <https://doi.org/10.1145/3473970>

This work is partially supported by the following fund: RGC#14204118 and RGC#RSFS3133237. Jing Li is supported by CCF-Tencent Rhino-Bird Young Faculty Open Research Fund (R-ZDCJ), NSFC Young Scientists Fund (62006203), PolyU internal funds (1-BE2W, 1-ZVRH, 1-ZVVX), and RGC Early Career Scheme (25200821).

Authors' addresses: X. Zeng, L. Wang, and K.-F. Wong, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China; emails: zxshamson@gmail.com, [lzwang,kfwong}@se.cuhk.edu.hk](mailto:{lzwang,kfwong}@se.cuhk.edu.hk); J. Li, Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China; email: jing-amelia.li@polyu.edu.hk.

This work is licensed under a Creative Commons Attribution-NonCommercial International 4.0 License.

© 2021 Copyright held by the owner/author(s).

1046-8188/2021/11-ART49 \$15.00

<https://doi.org/10.1145/3473970>

1 INTRODUCTION

While the exposure to fresh news is mediating through social media [8], online platforms have profoundly revolutionized the way people collect information and share viewpoints. Many individuals now turn to social media to voice opinions and discuss topics, which has expanded the traditional form of “kitchen table conversations” to public discussions in the online world. Especially under the crisis of COVID-19, online communication has become the only way for people in lockdown or quarantine to keep in touch. Meanwhile, the abundance of online content has presented the concrete challenge of information explosion, where massive online discussions formed every day hinder people’s capability to find the ones they want to engage in. Moreover, individuals have limited attention and different tastes, thus the vast bulk of social media information is usually irrelevant or unpalatable to them. For example, a big fan of President Trump might prefer a conversation concerning Trump’s issue, and only then if it is discussed in a positive light without controversy [1, 8].

Facing large quantities of superfluous materials, how can we help individuals to find a conversation to enter, and how might we automatically recommend online discussions to users that they would like to keep engaging in? To figure out the answers, we study the problem of *online conversation recommendation*, which aims to identify social media discussions fitting a user’s personal preferences, hence likely to result in their future engagement. Such an effective conversation recommendation engine will potentially provide a more positive social interaction experience to online users [17].

Nevertheless, most prior efforts made in this subject have focused on the information from individual posts for recommendation [18, 86, 95]. Others exploring conversation-level suggestion utilize handcrafted features [17] or word co-occurrence patterns [99], largely ignoring the useful information embedded in the local interaction structure formed with users replying to each other. Our previous attempt [101] captures reply structure and temporal features in local interactions, which presents state-of-the-art results. Compared with it, our model additionally captures word statistical patterns and allows the modeling of global word co-occurrence on conversation level and beyond a single turn. In doing so, we are also able to capture word distributions to reflect user preferences from both what they said (topical contents) and how they said (pragmatic behaviors), which thus produces more representative and interpretable representations.

To exploit interaction structure in users’ chatting history, we organize a conversation into a directed acyclic graph (henceforth *interaction graph*) whose vertices refer to conversation turns while edges denote turn interactions. Here, we consider two types of edges: (1) *replying edges* directed from a message to its reply to indicate in-reply-to relations; (2) *temporal edges* are used to link messages by their chronological order. Examples of interaction graphs are shown in Figure 1, where red arrows indicate reply edges and blue arrows denote temporal edges. Based on interaction graphs, we further learn participants’ preferences from what they discussed and how they interacted with each other, where *global and local interactions* are leveraged to recognize salient features from historical user behaviors. Here *local interactions* are learned from the edges of an interaction graph to reflect replying relations and chronological order. For *global interactions*, we capture word statistical patterns globally exhibiting in conversations to model two latent factors: (1) *topics* for representing user interests and conversation content and (2) *discourse modes* (such as *agreement*, *question*, and *emotional reactions*) for describing user interaction behavior and conversation dynamics.

Intuitively, neither global information nor local information alone is sufficient to provide reliable recommendations. For example, given two discussions both centering about football matches, a model that does not take the local information into account might be not able to

Fig. 1. Two Reddit conversation snippets on the right, both in form of their interaction graphs; red arrows indicate in-reply-to relations, and blue arrows depict chronological order. User U_0 is the target user for the later recommendation, 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). $[U]$ means that the message is posted by user U , where $U \in \{U_0, U_1, OU\}$ and OU is a general designation to other users. The italic words in the angle blankets $\langle \rangle$ are our interpretations of the messages' discourse modes, which can be indicated with the **discourse words** in blue. **Topic words** showing discussion topics are highlighted in red.

distinguish between them for a football fan. However, topical and discourse information that reflect user interests and behaviors are crucial for predicting users' future engagement, as birds of a feather flock together. To further illustrate how global and local interactions together could indicate users' future conversation behavior, we take the two Reddit conversation snippets shown in Figure 1 as an example. As can be seen, both Conversation 1 (C_1) and 2 (C_2) are centering around the **September 11 attack (9/11)**; also displayed (on the left) is the interaction history of the target user U_0 . We observe that user U_0 , who had discussed the event according to the chatting history, later engaged in C_1 instead of C_2 . Possible explanations could be (1) C_1 was initialized by user U_1 , who used to reply and comment on U_0 's viewpoints in prior discussions (reflected by *replying* relations in local interactions); (2) C_1 focused on U_0 's discussion interested *topic* "9/11," which can be drawn from the content words in their interaction history globally, such as "9/11" and "terrorists"; and (3) U_1 's argument with others (suggested by the "Negative Answer" discourse mode) drew U_0 's involvement, as they enjoyed the adversarial *discourse* in the debate with others.

In methodology design, our conversation recommendation framework is built on the success of **collaborative filtering (CF)**, which is originally proposed for product recommendation. It learns latent dimensions of product ratings or reviews to capture user preferences [33, 50, 58, 72, 88]. We follow neural-based CF framework [33]. It allows us to capture hidden factors that reflect user interests based on the conversation history joint with other modules in an end-to-end manner.

Furthermore, we learn semantic features from global and local interactions in users' conversation history as an understanding of user interests.¹ Here *local interactions* are encoded with graph-structured neural networks over interaction graphs to explicitly encode who replies to whom at when in the discussion history. For *global interactions*, probabilistic latent variable models are proposed to discover word distributions of latent topics and discourse modes in

¹To ensure the general applicability of our approach, we do not utilize external features such as social network structure [7].

ongoing conversations. To allow better collaboration with a neural-based framework, our global interaction modeling module learns latent topics and discourse modes based on **variational autoencoder (VAE)** networks [61, 98]. In this way, topic and discourse representation learning can be jointly trained with the CF framework in a more suitable manner. *To the best of our knowledge, our article is the first work to explore the joint effects of global and local interactions to predict how likely a user will engage in a conversation.*

To evaluate our models, we conduct extensive experiments on three datasets containing online conversations from Twitter and Reddit. The main comparison results with the previous conversation recommendation models show that our model leveraging both global and local interaction features presents the best performance, significantly outperforming the state-of-the-art model [101] that only learns local interactions from replying structure and temporal features. For example, our model achieves a **mean average precision (MAP)** of 0.570 on Reddit conversations compared with 0.538 presented by Zeng et al. [101]. We further find that our model still exhibits superior performance when the sparsity degrees of user history and conversation context is varied, which demonstrates our model's potential ability to handle sparsity in either conversation or user chatting records. Moreover, we investigate a challenging task to predict first-time replies, where our model again outperforms existing popular recommendation systems. Finally, we conduct detailed analyses over the global and local interactions learned by our model. The results show that our model can not only discover reasonable topic and discourse representations to characterize user interests and discourse behaviors but also captures important features from turn interactions in conversation structure. In doing so, we show how global and local interactions jointly indicate users' future activities, which interprets why it achieves advanced performance in conversation recommendation.

In summary, the contributions made in this article are threefold:

1. *Coupling global and local interactions to learn user conversation preferences.* We are the first to explore the joint effects of global and local interactions in modeling the features that indicate which conversation a user is likely to engage in, while previous efforts ignore either local [99] or global [101] interactions.
2. *Effectively incorporating topic and discourse into neural recommendation framework.* Our work takes advantage of the recent process in neural latent variable models [61, 98], which allows topic- and discourse-level user preference factors to be exploited together with a neural CF framework.
3. *An extensive empirical study on online conversation recommendation.* Our model exhibits significantly better results than competitive conversation recommendation models when evaluated on real-world conversation corpora from social media. We also analyze the representations captured from global and local interactions, which provides further insights on conversation recommendation.

2 RELATED WORK

This article is in line with online conversation modeling, user response prediction, and recommender systems. In the following, we briefly describe the previous work on these subjects.

2.1 Online Conversation Modeling

Social media has attracted increasing attention in digital communication research. Earlier researchers focus on post-level modeling and extract essential information, trends, or patterns via analyzing the contents of a single post [2, 94]. However, owing to the short and informal nature of social media messages, most natural language processing models will suffer from the problem of data sparsity [49]. To deal with that, recent studies have found the effectiveness to understand social media posts in conversation context, which are formed with users replying with each

other [9, 48, 106]; after all, social media platforms are originally built for people to communicate with each other [67, 98]. It consequently presents the increasing popularity of conversation-level understanding and analysis, centering around diverse applications, such as the prediction of anti-social behavior [23, 24, 102], future trajectory of a discussion [7, 22, 40], and other conversational properties [32, 63, 70].

Most of the aforementioned applications (including our conversation recommendation task presented in this article) are based on the understanding of discussion processes, which focus on how to encode conversation context and capture the essential features therein. Previous methods in this area are mostly designed in probabilistic graphical models and neural network architectures.

2.1.1 Probabilistic Graphical Models. These models aim to explore word statistics in the conversation collection and learn latent semantic structures (e.g., distributional word clusters) therein. It has shown effectiveness in analyzing the latent topics (in the form of word distributions) with topic models in **Latent Dirichlet Allocation (LDA)** [13] fashion. Despite their success on formal-written and context-rich documents, such as scientific papers and news articles, previous studies then find out that standard topic models, relying on the modeling of document-level word co-occurrences, may suffer from severe data sparsity in short and informal online messages and exhibit a compromised performance [35, 107]. Therefore, previous researchers have made substantial efforts in how to alleviate the data sparsity to better model social media content [4, 60, 107].

In the existing work, coupling conversation structure and graphical model have become a popular choice in social media text analysis [4, 53, 68]. For example, Louis and Cohen [53] proposes a grammar model for topic segmentation in a lengthy conversation process; Ritter et al. [68] discover discourse word distributions to represent latent dialogue acts in Twitter conversations; in Li et al. [49] and Zeng et al. [98], topic and discourse words are recognized to learn latent variables reflecting the salient content being discussed (topics) and the participants' pragmatic behavior (discourse). Our article is inspired by how they learn topic and discourse representations from conversations, while we additionally exploit them to capture personalized user interests for the recommendation.

2.1.2 Neural Models. Neural architecture has achieved promising results in learning deep semantic features from conversation contexts. In this line, neural sequence models based on **recurrent neural networks (RNN)** were one of the most popular choices, which encode conversation turns in chronological order [30, 76]. For example, Cheng et al. [21] employ an RNN language model to explore latent language factors that connect different pieces of context in Reddit discussions; Jiao et al. [40] design a two-layer **Bidirectional LSTM (BiLSTM)** [31] model to encode turn-level content and conversation-level turn interactions, where the learned representations are shown useful to find conversation killers. Nevertheless, researchers then pointed out that the existing models may suffer from the data sparsity issue in online conversations and hence proposed other features to handle the noisy social media data. Some of them propose to employ the latent topics—in the form of word distributions and learned by statistical topic models [13] and demonstrate their usefulness in alleviating the data sparsity. The latent topics are either pre-trained and leveraged as additional features [55] or modeled in a neural architecture to be jointly trained with other conversation understanding tasks [90]. Some studies found it helpful to encode participants' information in modeling the conversation process, such as user profiles [71], their social networks [82], or historical chatting messages [100]. Other strategies include the usage of neural language generation [16], pre-detected dialogue acts [80], and multi-task learning with other speaker role identification [54].

Actually, online conversations with multiple participants are of a more complex structure rather than a simple sequential structure. The replying relations [59], social interactions [103], speaker

roles [27], and even knowledge graphs [25] can be incorporated into context and make an online conversation graph-structured. As the graph neural networks [45, 77] develop and apply in many natural language processing tasks, they are also employed by recent researchers for conversation modeling. For example, Ghosal et al. [27] apply Graph Convolutional Networks [45, 57] for emotion recognition in conversations, while the same network is utilized in Wei et al. [92] to predict rumor and veracity online. Graph-state or tree LSTM [11, 77] is also found effective in modeling online conversations with replying structure [47, 97]. Our model also applies graph neural networks for conversation recommendation. More differently, we incorporate user embedding to make the model aware of personality differences, which is more appropriate for a personalized recommendation.

2.2 User Response Prediction

Works on user response prediction analyze the propagation patterns of social media content, to predict whether a given online content will receive desired responses or a given user will be interested in and respond to some online information. The former we name *General Prediction*, will be reviewed in Section 2.2.1. The latter is more like a recommendation perspective, which we summarized as *Personalized Prediction*, and will be introduced in Section 2.2.2.

2.2.1 General Prediction. Some researches on general prediction focus on forecasting message popularity, i.e., whether a given social media post will get sufficient attention (reposting, replying, or others) [6, 79]. Other works investigate the internal propagation and interactions within online platforms and leverage more context information for user behavior prediction [7, 71]. Researches in message popularity forecast focus on the extraction of handcrafted features related to their objectives, e.g., social and sentiment features [6], arrival patterns and timing features [7], group information [15], and so on. Some works also apply probabilistic graphical methods to model propagation patterns or participants' networks of social media platforms, to predict further response behaviors [12, 104].

Prediction tasks leveraging on context information that include conversation and user interaction structure, are mostly used to predict discussion trajectory. For example, Cheng et al. [22] and Jiao et al. [40] explore conversation content and structure, to predict the continuity of one conversation. Some researchers pay closer attention to antisocial behaviors during online communication. They try to predict whether the current given conversation context will lead to trolling [23], impoliteness [102], controversy [34], or other derailments of online discussions. Another task that is more related to this article, is the user's future engagement prediction over given conversation context. Backstrom et al. [7] and Zeng et al. [100] formulate the re-entry prediction task to predict user's continuity engagement, only based on conversation content or history messages but not personal interests.

2.2.2 Personalized Prediction. Personalized prediction is to predict a specific user's responses to given social media content. This type of prediction is usually formulated as a recommendation task. Extensive studies have worked on modeling social media user interests and recommend suitable posts for them [3, 18, 26]. Different strategies have been explored in recent years. An intuitive method is to extract some user-specific content or social features and then recommend based on ranking methods [26] or knowledge base [62]. Collaborative ranking methods [18, 36] are also popular in previous works, since we can recognize user interests from their online behaviors like thumbing up or reposting some posts, which is more like implicit feedback. There are also researches focusing on graph-based methods, such as diffusion-aware models [65] or models based on social networks [3, 95], as well as neural methods [96].

Nevertheless, the aforementioned works only consider the content of a single post, leaving out rich information from the discussion context. In this article, we focus on user interests in participating in an online conversation. This enables us to utilize rich context messages and conversation structure to improve performance. While some similar concerns [17] still are in statistical-level try.

2.3 Recommender Systems

Recommender systems are information filtering techniques that deal with information overflow problems. Recent advances achieved in online services make an individual more easily access online content, like music, product reviews, news and its comments, and so on. How to filtering out the irrelevant content and provide suitable ones in a personalized manner, becomes essential requirements for many online services [74].

2.3.1 Typical Recommender Systems. In a typical recommender system, the problems can be twofold: recommendation with explicit feedback and that with implicit feedback [10]. The former means that we can attain explicit user preferences while doing recommendations, like the ratings given to products. Such a problem can be evaluated by the differences between ground truths and predictions. While in recommendation with implicit feedback, user interests can only be estimated by some indirect behaviors, like purchase history or clickthrough history [37]. And it is usually formulated as Top-N ranking problems.

Typical methods for recommender systems mainly can be classified into three categories: collaborative filtering methods [75], content-based methods [84], and the hybrid of them [81]. Collaborative filtering methods are the most popular recommendation approaches, under the assumption that people who have similar interests before would prefer similarly in the future [73]. Researchers explore different ways to identify similar items or users, including memory-based methods [14], clustering methods [83], Bayesian models [78], and matrix factorization methods [46]. However, regarding those new items, which lack history information, CF-based methods may perform poorly. Content-based methods mainly compare the similarity between descriptive characteristics of items and user profiles, and recommend items that are similar to what users like before. Therefore, it is usually efficient in recommending new items but limited in terms of diversity and novelty [52]. As for hybrid methods, they take advantage of both CF-based methods and content-based methods [81]. Their shortcoming might be too complicated for some common problems.

2.3.2 Recommendation on Social Media. Social media platforms like Twitter, Reddit, or Facebook are also online services that attract a vast amount of users to share and discuss. With so much information posted by other individuals or organizations, recommendation techniques are also essential in social media. Apart from content recommendation introduced as personalized prediction in Section 2.2.2 (i.e., post or conversation recommendation), other recommendation tasks on social media are also extensively studied. For example, many efforts are paid on community recommendation, recommending friends or similar users for users to follow [5, 91], or interesting communities and groups to join [20, 105]. There are also works on mention recommendation, recommending usernames when people attempt to use the “@” sign to mention others [38, 87]. In Twitter-like social media, hashtag (keywords prefixed with the symbol “#”) or topic recommendation is also popular [28, 39].

Collaborative filtering frameworks are still the most popular methods utilized in social media recommendation [19, 20]. To alleviate the sparsity problems social media face and incorporate more available social information, more recent works attempt to use different models. For example, enhancing recommendation with topic models [28, 91], appending context or social features [19], and so on. Neural models like **convolutional neural networks (CNN)** [29], memory

networks [38, 56] also achieve attractive results. We examine conversation recommendation on social media in this article, exploiting both topic modeling and graph modeling as parts of our method.

3 CONVERSATION RECOMMENDATION MODEL

In this section, we propose our model based on global and local interactions for conversation recommendation. We first introduce the overall idea of our recommendation framework in Section 3.1. The next two subsections describe how we model local (Section 3.2) and global (Section 3.3) interactions with graph neural networks and neural latent variable models, respectively. Finally, we address how to incorporate them into the neural CF framework in Section 3.4, together with our training objective and joint training process.

3.1 Model Overview

We first describe the input and output. For training, our model is fed with a conversation dataset C . Each conversation $c \in C$ is formed with a sequence of turns $\langle t_{c,1}, t_{c,2}, \dots, t_{c,M_c} \rangle$, where M_c denotes the number of turns. Each turn t is in form of a word sequence $\langle w_{c,t,1}, w_{c,t,2}, \dots, w_{c,t,N_t} \rangle$ with N_t being t 's word number. Its author is represented by user id $u_{c,t}$. 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 structure within a conversation can be captured.

For recommendation, our model takes a user u and a conversation c as input, and then predicts how likely u will engage in c , conditioned on u 's previous behavior and c 's context history. Formally, our model outputs $\hat{y}_{u,c} \in [0, 1]$ for each user-conversation pair, which measures the probability that user u will engage in conversation c .

Our model is based on the neural-based **Latent Factor Model (LFM)**, a popular collaborative filtering framework [33, 75]. Basically, LFM assumes that there exists an unknown low-dimensional representation of users and items (i.e., conversations in conversation recommendation) where user-item affinity can be modeled accurately. In this work, to estimate $\hat{y}_{u,c}$, we extend the traditional LFM and divide the latent factors into two types: preference factors and interaction factors. Preference factors reflect conversation characteristics and users' corresponding preferences, while conversation structure and user interaction patterns can be represented by interaction factors. Therefore, $\hat{y}_{u,c}$ can be computed by the following equation:

$$\hat{y}_{u,c} = \lambda \left(\mathbf{r}_u^{PF} \otimes \mathbf{r}_c^{PF} \right) + (1 - \lambda) \left(\mathbf{r}_u^{IF} \oplus \mathbf{r}_c^{IF} \right) + bias, \quad (1)$$

where $\mathbf{r}_*^{PF}$ denotes preference factors, $\mathbf{r}_*^{IF}$ denotes interaction factors. λ controls the tradeoff between these two factors. " $\otimes$ " and " $\oplus$ " represent different operations to estimate similarity regarding the two kinds of factors (usually dot product in LFM). And $bias$ 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. The basic framework for our models is described in Figure 2. Next, we will introduce how we obtain the two different factors by learning local interactions (Section 3.2) and global interactions (Section 3.3) in conversations.

3.2 Learning Local Interactions

Local interactions reflect the internal interactions among participants within one conversation, which are implied by the replying and temporal relations of turns. To fully investigate their efficacy, we apply BiLSTM [31] and **Graph Convolutional Networks (GCN)** [45, 57] to model conversation interaction structure. These two networks are employed on the interaction graph formed by conversation turns, whose vertices refer to conversation turns and the edges denote the two kinds of turn interactions—replying edges and temporal edges. In the following, we first

Fig. 2. The basic architecture of our proposed conversation recommendation framework.

show how we obtain the turn-level representations. The detailed operation for modeling turn interactions is shown later on top of the turn representations. Finally, the extracted conversational representations are delivered for further operation.

3.2.1 Turn-level Modeling. Here we describe how we model turn-level representations. We first explore turn representations from how words appear therein, where we employ a CNN [42] 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 $I^W(\cdot)$ (initialized with pre-trained word vectors) to explore deep word semantics. 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 r_t^{TL} .

To encode personalized information, we also propose to incorporate author information into turn representations. Therefore, a user embedding layer $I_U^{IF}(\cdot)$ is employed, where the embedding vector r_u^{IF} for user u reflects u 's interaction patterns, such as what they used to say and whom they usually interacted with.

We concatenate r_t^{TL} , conveying content features, and $r_{u,c,t}^{IF}$, embedded with the interaction patterns of t 's author, to produce a turn representation r_t^{TR} . It couples turn t 's text content and its author's interaction patterns. We can summarize the operation as follows:

$$r_t^{TR} = [r_t^{TL}; r_{u,c,t}^{IF}], \quad r_t^{TL} = \text{CNN}(I^W(w_{c,t})), \quad (2)$$

where $[\cdot]$ indicates concatenation and $w_{c,t}$ represents the word sequence. Afterwards, r_t^{TR} is delivered to model t 's interaction with other turns in c .

3.2.2 Turn Interaction Modeling. To encode conversation interaction structure, the turns in a conversation c are organized 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, 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, $E^S(t)$. Then, we employ graph neural networks to model the interaction structure. We choose to use GCN [45, 57] (due to their better efficiency and performance, see Table 8) and describe how it models conversation interactions below.

Following previous practice [57], before using GCN to model inter-turn interactions, we first feed the turn representations r_t^{TR} into a sequential BiLSTM [31] layer to capture the chronological

interactions. With such a layer, we distinguish the temporal interactions from replying interactions and emphasize the effects of temporal features, which is essential on social media. Then, we take the t th hidden state of BiLSTM $\mathbf{h}_t^{LSTM}$ to further capture turn t 's interaction with its neighbors on interaction graph. The formula describing this process is given as

$$\begin{aligned} \mathbf{h}_t^{GCN} = & \sum_{i \in EP(t)} \omega_{i,t} (W^{pre} \mathbf{h}_i^{LSTM} + b^{pre}) \\ & + \sum_{j \in ES(t)} \omega_{j,t} (W^{suc} \mathbf{h}_j^{LSTM} + b^{suc}) \\ & + \omega_{t,t} (W^{self} \mathbf{h}_t^{LSTM} + b^{self}) \end{aligned} \quad (3)$$

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

$$\omega_{i,j} = \sigma \left(\mathbf{h}_i^{LSTM} \cdot V^{dir(i,j)} + d^{dir(i,j)} \right). \quad (4)$$

Such a gate is used 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 multi-layer GCNs, where we apply a ReLU activated function between two layers. In this way, multi-hop relations can be explored. The hidden states $\mathbf{h}_t^{GCN}$ after multi-layer GCNs serve as representations for local interactions and will be used to produce conversational features later together with global interactions.

3.3 Learning Global Interactions

Word statistical patterns globally exhibiting in conversations have been shown effective in capturing meaningful contextual representations for downstream tasks like classification [98], generation [89, 90], and so on. We refer to them as global interactions, since they are captured within the whole corpus and reflect global word-level distributions. In this work, we follow Zeng et al. [98] and focus on topics and discourse modes, which can be corresponded to preference and interaction factors mentioned in Equation (1). Specifically, to be consistent with local interaction modeling and enable joint learning, we employ a **neural topic model (NTM)** [61, 98] to discover word clusters reflecting what a conversation concerns (i.e., topics) and how participants voice their opinions (i.e., discourse modes) in an unsupervised manner.

3.3.1 Neural Model for Topics and Discourse. Our neural model that jointly learns topics and discourse modes (noted as TDM) is based on VAE framework [44] to simulate the corpus generative process. The VAE framework consists of an encoder and a decoder. The encoder maps a given input into latent topic and discourse representations, while the decoder reconstructs the original input from the latent representations.

For the encoder, we learn the parameters μ_T and σ_T , which are mean and variance of prior probability for topics, and π_D for discourse modes, from the input information. The input includes the Bag of Words (bow) representations of target turn t and its conversation context c , where we note as $\mathbf{t}_{bow}$ and $\mathbf{c}_{bow}$. The following formulae describe the details:

$$\begin{aligned} \mu_T &= f_\mu(f_e(\mathbf{c}_{bow})), \quad \sigma_T = f_\sigma(f_e(\mathbf{c}_{bow})) \\ \pi_D &= softmax(f_\pi(\mathbf{t}_{bow})), \end{aligned} \quad (5)$$

where $f_*(\cdot)$ represents a neural perceptron, with a linear transformation of inputs activated by a ReLU activation function.

As for the decoder, it reconstructs the words in target turn t with latent topic assignment $\mathbf{z}$ and discourse assignment $\mathbf{d}$ (drawn utilizing parameters learned from the encoder). The generative process of the reconstruction is follows:

- Draw the latent topic $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_T, \sigma_T^2)$, and compute $\mathbf{c}$'s topic distribution $\boldsymbol{\theta} = \text{softmax}(f_\theta(\mathbf{z}))$
- Draw the latent discourse mode $\mathbf{d} \sim \text{Multi}(\boldsymbol{\pi}_D)$
- For word index $n = 1$ to $N_{c,t}$:
 - compute distribution for word $w_{c,t,n}$: $\eta_n = \text{softmax}(f_{\phi^T}(\boldsymbol{\theta}) + f_{\phi^D}(\mathbf{d}))$
 - Draw word $w_{c,t,n} \sim \text{Multi}(\eta_n)$,

where the topic word distribution ϕ^T and discourse word distribution ϕ^D are obtained from the weight matrix of $f_{\phi^T}(\cdot)$ and $f_{\phi^D}(\cdot)$. Here at the corpus level, we assume there are K topics represented by word distribution ϕ_k^T ($k = 1, 2, \dots, K$), while ϕ_d^D ($d = 1, 2, \dots, D$) represents the D discourse modes embedded in corpus. K and D are hyper-parameters pre-defined.

With such process, we derive topic assignment variable $\mathbf{z}$ for conversation c and discourse assignment variable $\mathbf{d}_t$ for each turn t in c . They reflect the conversation's topical content and how the participants contribute to it, which are useful for conversational modeling together with local interactions.

3.3.2 Model Inference. Following Zeng et al. [98], the objectives for unsupervised model learning should consider the following three aspects: the learning of latent topics and discourse, the reconstruction of the target messages, and the separation of topic-associated words and discourse-related words. We describe them in turn as below.

Learning Topics and Discourse. To approximate posterior distribution over the topic variable $\mathbf{z}$ and the discourse variable $\mathbf{d}$ given all the training corpus, we maximize the variational lower bound for $\mathbf{z}$ and $\mathbf{d}$, defined as following:

$$\mathcal{L}_{zd} = \mathbb{E}_{q(\mathbf{z}|\mathbf{c})}[p(\mathbf{c}|\mathbf{z})] - D_{KL}(q(\mathbf{z}|\mathbf{c})||p(\mathbf{z})) + \mathbb{E}_{q(\mathbf{d}|\mathbf{t})}[p(\mathbf{t}|\mathbf{d})] - D_{KL}(q(\mathbf{d}|\mathbf{t})||p(\mathbf{d})), \quad (6)$$

where $q(\mathbf{z}|\mathbf{c})$ and $q(\mathbf{d}|\mathbf{t})$ are approximated posterior probabilities describing how the topic $\mathbf{z}$ and discourse $\mathbf{d}$ are generated from the corpus. $p(\mathbf{c}|\mathbf{z})$ and $p(\mathbf{t}|\mathbf{d})$ represent the corpus likelihoods conditioned on the latent variables. $p(\mathbf{z})$ follows the standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $p(\mathbf{d})$ is the uniform distribution $\text{Unif}(0, 1)$. D_{KL} refers to the Kullback–Leibler divergence that ensures the approximated posteriors to be close to the true ones.

Reconstructing Target Turns. The learning objective for reconstructing the target turn t from latent variables $\mathbf{z}$ and $\mathbf{d}$ is designed as

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{z}|\mathbf{c})q(\mathbf{d}|\mathbf{t})}[p(\mathbf{t}|\mathbf{z}, \mathbf{d})]. \quad (7)$$

Distinguishing Topics and Discourse. To let topics and discourse capture different information in a conversation, a mutual dependency between $\mathbf{z}$ and $\mathbf{d}$ is introduced here with the following equation:

$$\mathcal{L}_{MD} = \mathbb{E}_{q(\mathbf{z})}[D_{KL}(p(\mathbf{d}|\mathbf{z})||p(\mathbf{d}))]. \quad (8)$$

By minimizing $\mathcal{L}_{MD}$, we can guide the model to learn different word distributions for topics and discourse. More details about the above three losses can be referred to Zeng et al. [98].

Therefore, the overall objective for the whole TDM model will be as follows:

$$\mathcal{L}_{TDM} = \mathcal{L}_{zd} + \mathcal{L}_t - \rho \cdot \mathcal{L}_{MD}, \quad (9)$$

where ρ controls the tradeoff between $\mathcal{L}_{MD}$ and other losses in TDM module. By maximizing $\mathcal{L}_{TDM}$ via back propagation, the word distributions of topics as well as discourse modes can be jointly learned from conversations.

Fig. 3. The detailed architecture of our recommendation framework with global and local interactions, which models replying preferences and conversation interaction structure, as well as global statistical patterns, to predict a user's future engagement.

3.4 Recommendation with Global and Local Interactions

In this section, we introduce how we incorporate the representations from local interaction modeling (h_t^{GCN}) and global interaction modeling (z, d_t) into our recommendation module, which is based on neural CF model [33]. Typically, we need to extract preference factors and interaction factors both for users and conversations, so that we can later produce predicted score $\hat{y}_{u,c}$ using Equation (1). Figure 3 shows the detailed model architecture.

3.4.1 User Modeling. Our user modeling module provides user preference factor r_u^{PF} and user interaction factor r_u^{IF} for final prediction. To ensure the general applicability of our approach as described in Section 1, we do not utilize extra information like social networks. We extract user information merely from what they said and which conversations they have participated in. For that, we simply introduce two embedding layers $I_U^{PF}(\cdot)$ and $I_U^{IF}(\cdot)$ to encode each user ID into a preference hidden representation r_u^{PF} and an interaction hidden representation r_u^{IF} . And then the parameters of these two embedding layers are learned via the Equation (1) based on participated history in training data.

It is worth noting that we have introduced the same interaction embedding layer $I_U^{IF}(\cdot)$ in Section 3.2.1 to incorporate personalized information in each turn. In this way, our user modeling module also benefits from the local interaction modeling for each conversation. Moreover, this makes our model aware of the differences between r_u^{PF} and r_u^{IF} , so that r_u^{IF} is forced to encode information about interaction rather than user preferences.

3.4.2 Integration of Global and Local Interactions. To integrate global and local interactions and provide conversational factors for recommendation, we first follow previous work [99] and regard the topic variable z as the latent factor representing conversation characteristics, i.e., the preference factor r_c^{PF} . This is because z is obtained in conversation level and reflect topical content, which is very similar to the definition of preference factors in Section 3.1. We design two kinds of methods to achieve this goal. One is to apply a conversation embedding layer $I_C^{PF}(\cdot)$ initialized with z , where the corresponding embedding vector for conversation c is r_c^{PF} (used when we fix the TDM module in training); another is to directly make $r_c^{PF} = z$, which can be jointly trained in a multi-task manner.

For the interaction factor $\mathbf{r}_c^{IF}$, we simply concatenate the hidden representations reflecting local interactions $\mathbf{h}_t^{GCN}$ and the discourse variable $\mathbf{d}_t$ as the final turn representation for each turn, since $\mathbf{d}_t$ also reflects the interaction structure within a conversation. We then apply average pooling on them to produce $\mathbf{r}_c^{IF}$. We summarize them as follows:

$$\mathbf{r}_c^{IF} = \frac{1}{M_c} \sum_{t=1}^{M_c} [\mathbf{h}_t^{GCN}; \mathbf{d}_t], \quad (10)$$

where M_c is the number of turns in conversation c .

Finally, to encode different kinds of similarity between preference factors and interaction factors, we use different operations for “ $\otimes$ ” and “ $\oplus$ ” in Equation (1). Preference factors $\mathbf{r}_u^{PF}$ and $\mathbf{r}_c^{PF}$, which reflect user preferences toward different conversation characteristics, are estimated with dot product; while interaction factors $\mathbf{r}_u^{IF}$ and $\mathbf{r}_c^{IF}$ reflect user’s interaction patterns and conversation’s interaction structure, which might need deep exploration for their joint effects, so we apply **Multilayer Perceptron (MLP)** motivated by He et al. [33]. The following equations describe these two operations:

$$\mathbf{r}_u^{PF} \otimes \mathbf{r}_c^{PF} = \mathbf{r}_u^{PF} \cdot \mathbf{r}_c^{PF}, \quad (11)$$

$$\mathbf{r}_u^{IF} \oplus \mathbf{r}_c^{IF} = \alpha \left(W_M^T \left(\dots \alpha \left(W_1^T [\mathbf{r}_u^{IF}; \mathbf{r}_c^{IF}] + b_1 \right) \dots \right) + b_M \right), \quad (12)$$

where $\alpha(\cdot)$ is ReLU-activated function and M is the number of layers in MLP. With these two operations and Equation (1), the final predicted score $\hat{y}_{u,c}$ is derived for recommendation.

3.4.3 Joint Training Procedure. This subsection describes the training objective for our recommendation model, followed by several joint training procedures for the two modules (global and local interaction modeling) in our model.

Training objective. Our learning objective for recommendation is built on the success of one-class collaborative filtering [64], which weights positive instances higher during training to tackle implicit feedback on platforms like social media. Formally, we apply *weighted binary cross-entropy loss*, defined as follows:

$$\mathcal{L}_{BCE} = - \sum_{(u,c) \in \mathcal{T}} [\mu \cdot y_{u,c} \log(\hat{y}_{u,c}) + (1 - y_{u,c}) \log(1 - \hat{y}_{u,c})], \quad (13)$$

where $y_{u,c}$ is a binary label indicating whether u replied to c and $\hat{y}_{u,c}$ is our predicted score. $\mathcal{T}$ is a set of training instances consisting of several user-conversation pairs. μ ($\mu > 1$) is a pre-defined parameter to trade off the weights of positive and negative instances. During model training, the negative sampling strategy is adopted [33], whose sampling ratio (the number of negative samples for each positive instance) is set to 5.

For the whole model, the joint objective that needs to be minimized should consider both the recommendation objective and the objective for our TDM module, which is defined with the following equation:

$$\mathcal{L} = \mathcal{L}_{BCE} - \gamma \cdot \mathcal{L}_{TDM}. \quad (14)$$

Joint training procedure. To fully investigate how the global and local interactions can be learned together efficiently, we design three kinds of joint training procedures:

(1) We randomly initialize all the parameters of our joint model, and directly train the whole model with the objective of Equation (14) until convergence. We call the model trained with this procedure **RANDOM-CRGL**.

(2) We first pre-train our TDM module with the objective of Equation (9) to initialize our joint model, then the whole model is trained with the objective of Equation (14) until convergence. Such training procedure leads to model PRE-CRGL.

(3) We first train our TDM module with the objective of Equation (9) separately for our joint model, then the rest parameters are learned while fixing the TDM module. The process iterates several times until no more progress is made. The model trained with this procedure is referred to as Fix-CRGL.

The performance of these three models trained with different procedures will be first compared on the development set. The best model will serve as our final model for comparisons with other models as well as further analyses.

4 EXPERIMENTS ON CONVERSATION RECOMMENDATION

This section presents experiments on the conversation recommendation task with our proposed model. The datasets we use are described in Section 4.1. While in Section 4.2, we introduce how we set up the experiments, followed by the model setting, evaluation metrics, and compared methods. Finally, experiment results are showed in Section 4.3.

4.1 Datasets

To examine how our proposed model performs, three datasets from popular online platforms are used, where two of them are from Twitter, and the other one is from Reddit. Below, we describe how they are collected and preprocessed for experiments, along with the data structure.

4.1.1 Data Collection and Preprocessing. For Twitter data, we use the two conversation datasets released by Zeng et al. [99]: One contains discussions about the U.S. presidential election (henceforth USE), and the other gathers conversations of diverse topics based on the tweets released by TREC 2011 microblog track (henceforth TREC).² Tweets in USE were collected from January to June of 2016 with a small set of political keywords,³ while tweets in TREC dataset were mainly posted from January 23 to February 8, 2011. We also conduct the experiments on the Reddit conversation dataset used in Zeng et al. [101], which focuses on political issues (by choosing “politics” subreddit) from January to December 2008 (henceforth Reddit). To discover the whole conversations, messages were retrieved with replying relations (indicated by “*parent_id*” in Reddit corpus, for example). Their authors and parent messages were recorded in a recursive way until full conversations were recovered. Finally, conversations with only one message were removed.

For preprocessing, we follow previous work [99, 101] and apply the Glove tweet preprocessing toolkit [66]⁴ on the two Twitter datasets. For the Reddit dataset, we perform tokenization using the open-source natural language toolkit [51], 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 all datasets, including punctuation and emoticons.

4.1.2 Data Statistics and Analysis. The statistics of the three datasets are shown in Table 1. More details about the distributions of different aspects for the three datasets are displayed in Figure 4. We can observe that the three datasets exhibit different characteristics. USE dataset contains the fewest conversations, but conversations with longer turns account for a larger proportion. And users in it generally participate in fewer conversations (which makes it less user history context). The TREC dataset is a little larger, and users are spread across relatively more conversations (than

²<http://trec.nist.gov/data/tweets/>.

³Keyword list: “trump,” “hillary,” “clinton,” “president,” “politics,” and “election.”

⁴<https://nlp.stanford.edu/projects/glove/preprocess-twitter.rb>.

Table 1. Dataset Statistics

	USE	TREC	Reddit
Amount of users	4,300	10,122	13,134
Amount of conversations	2,013	7,500	29,477
Amount of turns	22,092	38,999	109,774
Average amount of message turns per user	4.6	3.9	8.4
Average amount of conversations per user	1.2	1.7	5.9
Average amount of turns per conversation	9.9	5.2	3.7
Average amount of users per conversation	2.6	2.3	2.6

Fig. 4. Distributions of different aspects for three datasets.

Fig. 5. Distributions for different conversation structures (left), with the new in rate and continue rate for different structures (right) on the three datasets.

USE). But most of the conversations in it only include two or three participants. In both Twitter datasets, user numbers are larger than conversation numbers, which shows that users on Twitter tend to discuss within a small circle. Reddit dataset contains more conversations than users, with users contribute more to different conversations. However, conversations in Reddit are generally shorter, which indicates that users like to participate in diverse and new conversations.

To further explore the effects of conversation structures implied by turn interactions, we display the distributions of different structures for the three datasets in Figure 5(a) and compute the new in rate (new users participate in that conversation) and continue rate (current participants continue to contribute in that conversation) for different structures in Figure 5(b). The path number means how many paths exist in the interaction graph from the root of conversation to the leaves, which can reflect the complexity of conversation structure. From Figure 5(a), most conversations

in all datasets are one-path, which means that the whole conversations are sequences following chronological order. More complex conversations are found in Reddit, which is in line with observations from Table 1. The effects of different structures can be found in Figure 5(b). In general, participants are more likely to continue in simple conversations (one-path), whereas conversations with complex structures (generally contain more users) tend to attract new users (observations are different in TREC dataset, that is mostly because complex conversations are rare in it, which can be observed from Figure 5(a)). Therefore, being aware of the conversation structure may be helpful for the recommendation.

4.2 Experimental Setup

In our experiments, we predict whether a user will engage in a conversation given the previous context messages in that conversation and past conversations before that. For each conversation, we take the first 75% of the context as observation for training purposes (at least one turn per conversation is retained for test or validation). 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.

4.2.1 Model Setting. Our model parameters are tuned on the development set, i.e., the parameters that give the lowest value for our objective (Equation (14)) are selected. Specifically, we set the tradeoff between preference and interaction factors (i.e., λ in Equation (1)) to 0.5, which implies that both factors are equally important. For confidence weight μ added for positive instances, it is set to be 100.

For the parameters regarding global interactions, the number of discourse modes (D) is 10, and the topic number (K) is 50. The tradeoff parameter γ between recommendation module and TDM module in Equation (14) takes value of 0.01. Another tradeoff parameter ρ balancing mutual dependency loss and other losses in Equation (9), is also set to be 0.01.

For the parameters with respect to local interactions, we initialize the word embedding layer with 200-dimensional Glove embedding [66], where the Twitter version is used for our Twitter dataset, and the Common Crawl version is applied on the Reddit dataset⁵ in turn-level modeling. For the CNN encoders, we use filter windows of 2, 3, and 4, each with 150 feature maps. In turn interaction modeling, the size of hidden states of our graph neural networks is set to 200, and we set the hidden size to 100 for each direction of BiLSTM.

Finally, the dimension for the interaction factors is 100, while the preference factors are with the same dimension as topic number K , i.e., 50. The number of MLP layers in Equation (12) is 3. During training, the batch size is set to 512 and Adam optimizer [43] is adopted with an initial learning rate of 0.01.

4.2.2 Evaluation Metrics. As a personalized recommendation task, for each given user in the test set, we produce a conversation list ranked by the predicted scores $\hat{y}_{u,c}$, and adopt popular information retrieval evaluation metrics to evaluate the performance. The final results are averaged over all users. The adopted metrics include MAP, **precision at n (P@n)**, and **normalized Discounted Cumulative Gain at k (nDCG@k)**. The formulae are as follows:

⁵<https://nlp.stanford.edu/projects/glove/>.

$$P@n = \frac{1}{|U|} \sum_{u=1}^{|U|} P@n_u = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{\sum_{i=1}^n R_u^i}{n}, \quad (15)$$

$$MAP = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{N} \sum_{n=1}^N P@n_u \times R_u^n, \quad (16)$$

$$nDCG@k = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{Z_u} \sum_{i=1}^k \frac{2^{R_u^i} - 1}{\log(1 + i)}, \quad (17)$$

where R_u^i is the goldstandard relevance score (1 if has been replied, 0 if not) for the i th conversation in the ranking list for user u , $|U|$ is the number of users, N is the number of conversations in produced ranking list, and Z_u is a normalization factor whose value is the maximum possible DCG through position k , also called Ideal DCG through that position.

4.2.3 Comparisons. For comparison, we first consider two simple baselines:

- **RANDOM:** ranking conversations randomly;
- **POPULARITY:** more popular conversations (i.e., with more distinct users) ranked higher;

Then, we compare with two traditional ranking models that are utilized on social media:

- **TOPICRANK** [17]: ranking conversations by topic relevance to the target user's historical messages, where topics are learned from both LDA [13] and TF-IDF statistics.
- **RANKSVM** [26]: ranking conversations for each user based on some stated features, with ranking SVM [41].

We further compare some relative non-neural models:

- **CTR** [18]: a CF-based social media post recommendation model, where we aggregate messages in one conversation as one large post to fit in the setting.
- **HFT** [58]: a model also leverages topic modeling techniques to enhance recommendation performance.
- **CRJTD** [99]: a recommendation model that only considers global interactions (i.e., topics and discourse).

Last, we compare with neural recommendation models:

- **NCF** [33]: the neural CF model, not utilizing any context information.
- **HUITA** [93]: a CNN-based model, where a hierarchical attended CNN framework is exploited to model both user and conversation structure.
- **CRIM** [101]: A model that only considers local interactions under the similar neural CF framework and holds state-of-the-art results.

For our proposed model, we first compare the performance with different training procedures, i.e., the performance of **RANDOM-CRGL**, **PRE-CRGL**, and **FIX-CRGL** (Section 3.4.3). Then the model with the best results will be the final model to be compared with other models listed above.

4.3 Results

We display the main comparison results in this section. We first compare the performance of our model with different training procedures and determine our final model as the one with the best performance. We then compare it with other recommendation models, followed by three different analyses on model performance regarding different training context, different user sparsity, and first-time replies prediction.

Table 2. Results When Trained with Different Training Procedures

Models	Total Train Time	Train Time/Epoch	Dev. MAP		
			USE	TREC	Reddit
RANDOM-CRGL	1.04	1.25	0.710	0.564	0.514
PRE-CRGL	0.88	1.25	0.721	0.589	0.530
FIX-CRGL	1.00	1.00	0.737	0.618	0.563

“Total Train Time”: the total time used to train the model until convergence divided by that of Fix-CRGL. “Train Time/Epoch”: training time per epoch divided by that of Fix-CRGL.

Table 3. Main Comparison Results on Conversation Recommendation

Models	USE			TREC			Reddit		
	MAP	P@1	nDCG	MAP	P@1	nDCG	MAP	P@1	nDCG
Simple Baselines									
RANDOM	0.018	0.004	0.009	0.006	0.001	0.002	0.040	0.010	0.022
POPULARITY	0.050	0.010	0.025	0.023	0.005	0.010	0.082	0.033	0.063
Ranking Models									
TOPICRANK	0.666	0.668	0.670	0.454	0.512	0.471	0.351	0.523	0.418
RANKSVM	0.687	0.680	0.690	0.554	0.575	0.559	0.453	0.457	0.466
Non-neural Models									
CTR	0.673	0.649	0.678	0.475	0.431	0.495	0.278	0.178	0.276
HFT	0.698	0.652	0.706	0.487	0.447	0.504	0.436	0.542	0.470
CRJTD	0.762	0.750	0.757	0.591	0.591	0.600	0.453	0.559	0.485
Neural Models									
NCF	0.715	0.706	0.711	0.573	0.593	0.576	0.412	0.544	0.461
HUITA	0.716	0.709	0.713	0.580	0.596	0.585	0.506	0.590	0.549
CRIM	0.734	0.718	0.731	0.625	0.632	0.626	0.538	0.674	0.590
Ours									
OUR MODEL	0.846	0.833	0.847	0.634	0.649	0.637	0.570	0.703	0.616

“nDCG” stands for “nDCG@5.” The best result for each column is in **bold**. Our model significantly outperforms all comparisons from previous works ($p < 0.01$, paired t test).

4.3.1 Comparison on Training Procedures. We compare our models trained with the three designed training procedures and display the results in Table 2. As can be seen, Fix-CRGL achieves the best MAP scores on the development sets for all three datasets. Comparing RANDOM-CRGL and PRE-CRGL, we can find that our model converges much quicker and gets better performance with pre-training when jointly training the whole model. However, although Fix-CRGL needs less time for each epoch (since the TDM module is fixed during training the rest parameters), it still requires more total training time than PRE-CRGL, very close to RANDOM-CRGL. This is because that it needs more epochs for convergence and therefore achieves better optimum than the other two methods. We attribute it to the natural difference between the TDM module and other modules like BiLSTM and GCN. It might be very difficult for the model to achieve better optimum when jointly training with the objectives of recommendation (Equation (13)) and target reconstruction (Equation (9)). Therefore, our Fix-CRGL model, which trains them in a separated manner, can achieve better performance than the other two methods. In the rest experiments in this article, we will use Fix-CRGL as our final model for further comparisons and analyses.

4.3.2 Comparison with Previous Methods. Table 3 displays the main results of our model, compared with previous recommendation methods. We adopt MAP, P@1, and nDCG@5 here for

comparisons (we also try other metrics including $P@5$ and $nDCG@10$, and find similar trends). We can find our model achieves the best results on all the three datasets, showing the effects of both global and local interactions and the efficiency of our designed mechanism for utilizing both information. Below describes the more detailed observations.

First, we observe that both baseline models work poorly, implying that conversation recommendation is challenging, not possible to be well tackled with simple ranking strategies. While POPULARITY improves a little than RANDOM, it shows that hot spots (discussed by more users) are more likely to attract users. Two ranking models perform much better, which illustrates that domain-specific features are essential for the recommendation task, where the recommended items (conversations) contains lots of information. RANKSVM shows much better performance against TOPICRANK, that is because the features extracted following Duan et al. [26] are based on social media scenarios. Feature-based ranking models depend a lot on whether the extracted features can reflect the necessary characteristics. The results show that recommendation on social media may share some effective features.

CTR does not utilize turn-level information, resulting in its worse results than HFT and HUITA. These three models all do not consider conversation-level interactions, so that they are beaten by our model. As for NCF model, although it do not utilize any text information, it still produces competitive results in USE and TREC. This is because in Twitter, users tend to continue in their previous discussions (can be drawn from the continue rates showed in Figure 5(b)), resulting in better predictions even only with replying history. The worse performance on Reddit also proves the speculation.

Both global and local interactions are effective for conversation recommendation, implied by the better performance derived by CRJTD, CRIM and our model. Specifically, local interactions captured by turn interactions contribute more for those conversations with complex structure, indicated by much better performance achieved by CRIM than CRJTD on Reddit. CRJTD performs better than CRIM on USE dataset, showing that longer conversations with more focused topics benefit more from global information. As a result of that, our model achieves best performance, thanks to its ability of integrating both interactions. Next, we explore in-depth to show performance of our model in different situations.

4.3.3 Results with Different Training Context. The main results are reported given the first 75% context turns as conversation history. Here we investigate how the length of conversation training context affects reply preference prediction. The models are also trained with the first 25% and 50% turns as training context and their MAP scores on dataset USE and Reddit are showed in Figure 6.

We can first observe that all models perform worse when training with less context, which is in line with intuition that models need more context information to capture conversation characteristics. Another observation is, models considering global interactions (CRJTD and our model) drop less than those only with local interactions (CRIM) as decreasing training context. This is mainly ascribed to local interactions might need more turns to reflect their interaction cascades. However, global interactions, considering word distributions among the whole corpus, are affected less by context turn numbers. Our model performs much better and drops less for USE dataset, mainly resulting from longer average turn numbers in USE (9.9 from Table 1), where both global and local interactions still take considerable effects.

4.3.4 Results with Different User Sparsity. Table 1 and Figure 4(a) has shown that most users only engage in very few of conversations. This results in severe data sparsity in user interaction history, especially on Twitter. To study the effects of such data sparsity on recommendation model, we experiment and compare our model with the other methods regarding users with different

Fig. 6. MAP scores of models trained with the first 25%, 50%, and 75% turns as training context.

Fig. 7. MAP scores of models with different levels of user history sparsity on USE and TREC.

levels of interaction history (indicated by the numbers of conversations the user participated in training data). Figure 7 shows the MAP scores on dataset USE and TREC.

In general, all models perform much better with sufficient user interaction history (participated conversation numbers ≥ 2). CRIM and our model yield better results for users with high interaction sparsity, mostly because the local interactions captured by conversation interaction structure can help formulate user's interaction patterns, and so capture more user information. CRJTD performs a little better for users with more interaction history, indicating that global interactions are more helpful when we have sufficient history information.

4.3.5 Results for First-Time Replies. From a recommendation perspective, users might be interested in joining new conversations. We thus compare our model with other recommendation methods for prediction on first-time replies. For each user, we only evaluate considering conversations where they are newcomers.

Table 4. Results for First-time Replies Prediction

Models	USE			TREC			Reddit		
	MAP	P@1	nDCG	MAP	P@1	nDCG	MAP	P@1	nDCG
RANKSVM	0.023	0.007	0.013	0.002	0.000	0.001	0.049	0.027	0.036
HFT	0.054	0.017	0.036	0.058	0.041	0.059	0.077	0.038	0.053
CRJTD	0.083	0.053	0.067	0.090	0.076	0.094	0.088	0.045	0.065
HUITA	0.074	0.042	0.060	0.063	0.048	0.064	0.211	0.201	0.200
CRIM	0.139	0.079	0.123	0.160	0.138	0.156	0.212	0.202	0.203
OUR MODEL	0.383	0.305	0.403	0.160	0.131	0.159	0.244	0.213	0.225

"nDCG" stands for "nDCG@5." The best result for each column is in **bold**.

Table 4 shows the results. Unsurprisingly, all systems perform poorly on this task, though our model, as well as CRIM performs much better. This suggests that recommending unseen conversations is extremely difficult for the datasets with high sparsity problem. However, the local interaction information captured by replying relations is helpful on supplementing knowledge for recommendation, derived by the better performance of CRIM and our model. For this part, more personalized information like user profiles or social networks might help. However, these extra features might be domain-specific and how to better incorporate them deserves investigation. We will leave it as our future work.

5 ANALYSIS ON GLOBAL INTERACTIONS

In this section, we explore deeper into the learned representations (latent topics and discourse) of our model, which reflect the global interactions, to show whether our model captures meaningful latent representations. We first examine the learned latent topic and discourse variables respectively in Section 5.1 and Section 5.2. Then an ablation study is shown in Section 5.3 to indicate joint effects of latent topics and discourse. Finally, in Section 5.4, we also analyze the errors caused by our model, which may benefit the future improvement.

5.1 Latent Topics

We first examine the latent topics learned by our proposed model. To show the ability of the model learning effective topic distributions over the whole conversation corpus, we present the topic coherence scores, which can evaluate the quality of learned topics. We will compare with some topic modeling baselines. Then we also qualitatively analyze the interpretability of our learned topics via analyzing the word distributions of some sample topics.

5.1.1 Topic Coherence. The quality of topics derived from topic models, is commonly measured by topic coherence scores, assuming that words representing a coherent topic are likely to co-occur within the same document. We adopt the C_v coherence scores measured via the open-source Palmetto toolkit⁶ as our evaluation metric [69]. Table 5 presents topic coherence scores of the topics produced by our model, compared with baselines LDA [13], HFT [58], and CRJTD [99], selecting the top 5 and top 10 topic words as evaluation.

We can observe that both our model and CRJTD perform better than the other two baselines. A very important reason is that these two models learn word distributions based on both topics and discourse, which enables them to capture more explicit conversation topics. However, our model beats CRJTD in the all three datasets, indicating the end-to-end training manner in neural topic models also benefit in capturing topic information.

⁶<https://github.com/dice-group/Palmetto>.

Table 5. C_v Coherence Scores for Topics Produced by Various Models on Three Datasets

N	Models	USE	TREC	Reddit
5	LDA	0.421	0.441	0.467
	HFT	0.482	0.484	0.463
	CRJTD	0.512	0.510	0.477
	OUR MODEL	0.532	0.536	0.516
10	LDA	0.334	0.328	0.330
	HFT	0.365	0.386	0.343
	CRJTD	0.385	0.389	0.383
	OUR MODEL	0.404	0.427	0.431

N = the number of top words ranked by topic-word probabilities. Higher scores indicate better coherence. Best result for each setting is in **bold**.

Table 6. Top 10 Words on USE Dataset for Topic Relative to “Trump”

<u>Baselines</u>	
LDA	trump terrorist donald <i>say kill</i> people isis muslims <i>wrong</i> country
HFT	trump <i>care eye lol</i> racist right black support <i>kill</i> corbyn
CRJTD	<i>email</i> trump people <i>today understand</i> gop man white member win
<u>Ours</u>	
OUR MODEL	trump white kai win vote <i>let lrb kiss</i> spin corbyn

The words in *red italic* indicate non-topic words.

5.1.2 Sample Latent Topics. We also present a qualitative study on the produced topics of our model. Table 6 displays the top 10 words for topic “Trump” induced by our model, CRJTD only with global interactions and two topic modeling baselines on USE dataset (if there are multiple topics relative to “Trump,” then we pick up the most relevant one).

We can observe that it is challenging for the models to extract exactly representative topic words from social media messages, where the used languages are short and informal. All models may present some non-topic words in their top 10 words. For example, all models may produce irrelevant verbs, like “say” in LDA, “kill” in LDA and HFT, “understand” in CRJTD, and “let” in our model. This might be ascribe to such verbs frequently appeared in those messages about “Trump,” but the models lack ability to distinguish verbs and topic words (usually nouns). We can also find that CRJTD presents more topic-relative words in top 10 words rather than top 5 words. While our model is opposite, presenting more topic-relative words in top 5 words. The reason might be CRJTD learns topical features under the joint training framework, which also produce topic information from user behaviors, making the learned topics not focused. Our model learns topics separately, resulting the learned topics more focused on top 5 words.

5.2 Discourse Interpretation

The learning of discourse modes in both our model and CRJTD is under unsupervised mechanisms, which lacks sufficient precise labels. Therefore, we also present a qualitative study by presenting some sample discourse modes to show the reasonability of our learned discourse information. Table 7 shows a sample of discourse modes learned by our model and CRJTD on dataset USE and TREC. Although this is merely a qualitative human judgment at this point, there does appear to be interesting word clusters that reflect varying discourse modes found by our model, as well as CRJTD, without the guidance from any manual annotation on discourse.

Table 7. Top 10 Representative Terms for Sample Discourse Modes Discovered by CRJTD and Our Model in Datasets USE and TREC

Discourse	CRJTD		OUR MODEL	
	USE	TREC	USE	TREC
<i>Question</i>	? it if ... so all how because when any	? : <HASHTAG> what or too with <USER> and ...	<rsb> ask she “ ? <lsb> mmm family question because	? what how do you <REPEAT> why is does the
<i>Reaction</i>	you like any good ! please no ~ lol what	all thank haha <EMOTICAN> & !!! right ok u <SMILE>	<EMOTICAN> moan softly <lsb> kiss cheek gently thrust lol	<EMOTICAN> xx i <SMILE> <SADFACE> they but <LOLFACE> it <NEUTRAL- FACE>
<i>Statement</i>	's do think the . should they from and have	i , a what all you be how then ...	<rrb> indeed the agree i fully we remain share ,	is the <REPEAT> it ! i that in this of
<i>Argument</i>	but that all fu*k without ! against out though anything	do would up that too even always never anything much	you <rrb> explain yourself <lrb> read argument game ? democracy	is the <REPEAT> it ! i that how <EMOTICAN> but

Names of discourse modes are our interpretations according to the word distributions generated by our models.

By examining the internal word distributions within each discourse assignment, we find that both models can discover some general discourse modes on different datasets. We choose four discourse modes that are more obvious and interpretable and display the top 10 terms with our interpretation on the left (in Table 7) for their representative roles. We discuss each displayed discourse mode in turn as follows:

Question: expresses users asking questions to other users, mainly implied by the question mark “?”, “what,” “how” and other interrogative pronouns.

Reaction: reflects users’ emotions or feelings, indicated by emoticans, words expressing emotions like “lol,” “haha,” and so on.

Statement: presents users’ description or explanation with peaceful tones, generally reflected by “.” or “,” and words like “need,” “is,” “if” that describes some facts.

Argument: expresses users’ opinions by doubting, opposing others, or in a rude manner, which can be learned by bad languages like “fu*k,” or some negative words like “but,” “never,” and so on.

5.3 Balance between Topics and Discourse

We have shown that our model captures meaningful representations for latent topics and discourse above. Here we also provide an ablation study to examine the relative contributions of the two aspects, i.e., topics and discourse. To simplify the experiments, we work on the two variants of our model. In the first variant, we remove the initialization from topic variable z (noted as discourse-only); while we do not concatenate the discourse variables d_t with hidden states in Equation (10) for the second variant (noted as topic-only).

The results show that the topic-only model produces MAP scores of 0.783 and 0.552 on datasets USE and Reddit, while the MAP scores for the discourse-only model are 0.802 and 0.547, respectively. It indicates that both topic and discourse representations help on better recommendation, since we can make the learned conversation factors more representative. A similar ablation study has also been conducted in Zeng et al. [99]. They found that both topics and discourse are of the same importance in recommendation scenarios, which is consistent with our results.

5.4 Error Analysis

We further analyze the errors that might occur in our model, regarding the learned topics and discourse. For topics, we have briefly pointed out in Section 5.1.2 that our model might learn some non-topic words during topic learning. Specifically, the model mostly confuses topic words with some general verbs or sentiment words. For example, within the top 10 words produced by our model for topic “Trump” in USE dataset (Table 6), “let” is a general verb that does not reflect topic-relative information, but may frequently appear in those conversations. Another example is the top 10 words derived for the topic “Muslim” in Reddit dataset, which are “muslim reasonably canada wrong power organization policy obama isis argument.” We can find that it also contains sentiment words like “reasonably” and “wrong.” Frequent co-occurrences with topic words might be the main reason why our model produces such errors. Therefore, future work can focus on the separation of topic words and these irrelevant words. For example, incorporating POS tagging techniques, since a large portion of topic words may be nouns.

For discourse, the main problem lays in the interpretation potential of the found discourse clusters. In Zeng et al. [98], the authors pointed out that the NTM model that jointly learns topics and discourse may induce some discourse modes beyond the pre-defined labels. Although we do not have annotated labels in our datasets, we also find some discourse clusters that are hard for us to distinguish. For example, the top 10 terms derived from our model for one specific discourse cluster in TREC dataset are “<SMILE> <EMOTICAN> your the ? ! why screen what on.” It is more like a mixture of discourse *Question* and *Reaction*, but not exactly either of them. We can also observe that it contains the topic word “screen” by mistake, mainly resulting from the failure of separating topic and discourse words. Both errors show that our model still has the potential for improvement on separating different types of words. Another problem is that it is hard for our models to capture complex discourse behaviors, like irony. That is because our models detect latent discourse purely depend on the co-occurrence patterns of different words, while distinguishing those complex discourse behaviors might need deeper language understanding, which is beyond the capacity of our model.

6 ANALYSIS ON LOCAL INTERACTIONS

In this section, we analyze the internal representations of local interactions learned from our model to show that whether our model can learn reasonable local interaction features and show their benefit for the recommendation. For the following parts of this section, we first compare some alternative methods for turn encoder and interaction modeling, which is described in Section 6.1, to show the reasons why we choose the current modeling method. Then, in Section 6.2, we analyze the performance when our model deals with different conversation structures, followed by a sub-task considering turn-level prediction leveraging on our learned representations in Section 6.3. Finally, we also present an error analysis in Section 6.4.

6.1 Performance for Different Interaction Modeling Variants

To show why we choose the current method for local interaction modeling (i.e., use CNN as turn encoder, BiLSTM+CNN to model turn interaction, see Section 3.2.1 and 3.2.2), we display the

Table 8. Results for Variants That Use Different Turn Interaction Modeling Methods and Turn Encoders

Turn Interaction	Turn Encoder	Train Time/Epoch	Num. Params	MAP		
				USE	TREC	Reddit
BiLSTM+GCN	CNN	1.00	4.72M	0.734	0.625	0.538
	BiLSTM	1.20	4.75M	0.724	0.613	0.520
	Transformer	3.52	8.18M	0.766	0.642	0.556
BiLSTM	CNN	0.94	4.68M	0.724	0.588	0.492
GLSTM		1.25	5.28M	0.721	0.587	0.530
GCN		1.03	4.65M	0.721	0.589	0.534
Transformer	Transformer	5.41	12.34M	0.760	0.640	0.560

The first row is how we use in our final model. “Train Time/Epoch”: training time per epoch divided by that of our final model. “Num. Params”: the total number of trainable parameters. The best result for each dataset is in **bold**. The hyper parameters (including the number of model layers) of all variants have been tuned with MAP scores on development set, except for Transformer. We use at most four layers of Transformer, each with four attention heads, to save computing resources.

performance of different variants in Table 8, together with their training time and total numbers of parameters (to simplify and save training time, here we do not incorporate TDM module and merely show the performance with different kinds of local interaction modeling methods).

We first compare three kinds of turn encoders, CNN [42], BiLSTM [31], and Transformer [85]. Their comparisons are shown in the first three rows. As can be seen, CNN performs consistently better than BiLSTM, with similar model size and less training time. Therefore, CNN is chosen rather than BiLSTM. As for the Transformer, although it yields better results than CNN, the total model size is almost twice that of CNN, with more than three times of training time. Since the main concern in this article focuses on the exploration of global and local interactions for conversation recommendation, rather than selecting a better way to model text content, we use CNN instead of Transformer as turn encoder in our main experiments to save computing resources and training time.

We also compare different methods for turn interaction modeling, which is the main module to capture local interaction features. BiLSTM+GCN represents what we use in our model. We also examine only using BiLSTM or GCN layers for modeling, as well as replacing GCN with another kind of graph neural networks: **Graph-state LSTM (GLSTM)** [11, 77]. We can observe that BiLSTM is trained fastest due to its simple structure. But it performs much worse on Reddit dataset, attributed to the rich replying structure therein (as shown in Figure 5(a)). Sequential models cannot capture such graph structure. Graph neural networks GLSTM and GCN improve a lot on Reddit but suffer from more training time, especially for GLSTM. The best performance among them is achieved by BiLSTM+GCN, with less training time. This shows the effectiveness and efficiency to explore the order of turns with BiLSTM and the turn interactions with GCN.

At last, we explore using Transformer layers to encode turn interactions. In this way, the interactions among different turns are learned by themselves with self-attention rather than with replying or timing relations. It is interesting that it still yields competitive results compared to BiLSTM+GCN (see row 3 and row 7 in Table 8, both use Transformer as turn encoder). This indicates that turn content itself might have sufficient ability to learn local interactions, without any graph information. However, the price is more training time with a larger model (i.e., more computing resources). As replying relations and sequential features are common in different platforms and easy to retain, we prefer to encode local interactions with graph networks in this article to achieve better efficiency.

6.2 Performance at Different Interaction Structure

Local interactions are indicated by chronological order and replying relations among turns in one conversation. Generally, more participants might make the interaction structure (as an

Fig. 8. MAP scores of models regarding different conversation structure on Reddit.

Table 9. Accuracy for Turn-level Prediction Task

Models	USE	TREC	Reddit
RANDOM	0.076	0.145	0.212
CRIM	0.129	0.232	0.386
OUR MODEL	0.203	0.601	0.257

The best result for each dataset is in **bold**.

interaction graph) more complex and therefore need more efficient modeling. Here we would like to examine the performance of our model when dealing with conversations with different interaction structure, to show the efficiency of our interaction modeling framework. With such purpose, we separate our target conversations into two groups, categorized by their conversation structure (One-path and More-path, same as Figure 5). Then we can get a ranking list of users for each conversation based on the predicted scores $\hat{y}_{u,c}$ from our model. Finally, we produce MAP scores averaged over all conversations in that group. The results on Reddit dataset are presented in Figure 8. Here for comparison, we also display the results of our model removing the TDM module (denoted as “Ours-GI”) and further without GCN layers capturing replying relations (denoted as “Ours-GI-GCN”).

From the results, we can first observe that with GCN layers, our model can perform better both for one-path and more-path conversations, owing to the ability of GCN on modeling conversation structure. Our model and “Ours-GI” model produce higher MAP scores on more-path conversations than one-path ones. Without GCN, the model performs even a little worse on more-path conversations, since only a BiLSTM layer cannot capture exactly the structure of more-path conversations. Such results validate the effects of local interactions.

6.3 Turn-level Prediction: Effects of Learned Turn Representations

To further examine whether we learn meaningful representations for each turn in a conversation with our interaction modeling, we design an experiment to produce turn-level prediction with the learned turn representations. We first collect all the positive samples of user-conversation pairs (i.e., $y_{u,c} = 1$) from test set. And then find out which turn the user u finally replied to, setting that turn as true label 1, while other turns set to 0. We next compute the similarity between target user u and each turn, using cosine similarity over $\mathbf{r}_u^{IF}$ and turn representations ($[\mathbf{h}_t^{GCN}; \mathbf{d}_t]$ in Section 3.4.2, after linear transformation to the same dimension). Finally, we predict that user u is most likely to reply to the turn with the highest similarity. We present the accuracy results of such prediction in Table 9, compared with a baseline of random guess and similar operation on the outputs of model CRIM.

Fig. 9. MAP scores of our models on conversations with different lengths and different structures.

From the results, both our model and CRIM yield much better accuracy scores than a random guess, mainly attributed to our successful modeling for turn interactions, which proves the effects of our local interaction modeling module. However, we can also observe that datasets with longer conversations are more difficult for such prediction, since all methods perform worst on USE, in which average turn number achieves 9.9 (Table 1). Our model and CRIM perform differently on TREC and Reddit datasets, where our model achieves much higher scores on TREC, but opposite observations can be found on Reddit. The reason can be the difference between the two datasets. Conversations in TREC are mainly with simple structure (Figure 5(a)), which might be not useful to model with graph networks, while topics and discourse information derived from global interactions might help. But Reddit conversations are much shorter, the learned discourse information (also counted as part of turn representations in our model) might be less representative and therefore harm the performance.

6.4 Error Analysis

In this section, we analyze the insufficiencies of our model regarding modeling local interactions. First, the main reason that we decide to utilize graph neural networks modeling turn interactions is that we assume online multi-participant conversations are more likely to be in a complex graph structure. However, actually more than 70% of conversations in our datasets are of simple structure (Figure 5(a)), indicating the fact that maybe a linear model is sufficient for most conversations. The results from Section 6.1 and Section 6.2 prove the point to some extent, where only a BiLSTM can produce similar MAP scores even with less training time. To overcome such a situation, exploiting different modules for different conversations is a more effective strategy, but it might need more effort on deciding when to use which modules. An alternative method is to use Transformer for all cases, with the drawback of lower speed (Table 8).

However, our model also cannot handle some specific conversations well. To illustrate this, we compare the performance of our model regarding different conversations. Figure 9 shows the MAP score trends of our model concerning conversation lengths (Figure 9(a), USE dataset is selected due to its higher averaged turn number) and conversation structures (Figure 9(b), Reddit dataset with more complicated conversations is selected), as well as that of CRIM. We can find that both models achieve the best MAP scores in the middle of their curves, suffering performance drop when conversations are too long or too complicated. Lack of treatment on the sparse adjacency matrix derived from conversation structure might be the main reason. For conversations with a large number of turns, the adjacency matrix constructed using replying relations can be extremely sparse. Without dealing with such sparsity, models may derail later. Some methods can be applied to the original adjacency matrix to alleviate the effect. For example, Wei et al. [92] apply multiplication operation to expand the receptive field of each turn and add an identity matrix to increase self-importance.

In Section 6.3, we design a turn-level prediction task to evaluate the learned turn representations. It is observed that our model outperforms a random guess, indicating that the learned turn representations can represent some features. However, how useful are the representations? We design another baseline that always predicts that users will reply to the last turn, come from the intuition that the last turn reflecting the latest progress of that conversation might be more attractive. Sadly, such a baseline produces an averaged accuracy score of 0.4 for our datasets, which beats both the displayed models. This is because that our models are trained based on the conversation-level prediction, where we only leverage the averaged turn representations as conversation features for further computation. To make our model learn more meaningful turn representations, one possible way is to utilize multi-task learning techniques, which allows the model to jointly learn representations with a turn-level prediction task, drawing more compact representations.

7 CONCLUSION AND FUTURE WORK

In this article, we have presented the effects of global and local interactions for online conversation recommendation, where a powerful model is proposed for elaboration. Our model first uses graph neural networks to encode local interactions, where a Graph Convolutional Network following a BiLSTM layer is applied on top of turn content features. We then consider a neural topic model to discover topical and discourse distributions over the whole corpus as global interaction representations. Finally, the two modules are integrated into a neural collaborative filtering framework for further prediction.

Experiments on three datasets, from different platforms including Twitter and Reddit, show that our model performs better than the previous recommendation models, including previous models that only consider global or local interactions. It indicates that both global and local interactions are effective in capturing useful information for conversation recommendation. We also conduct extensive experiments to analyze the different aspects of our proposed model. The in-depth analyses illustrate that our model learns meaningful global and local interaction representations, which can explain the superiority of our model framework.

Our future work has two directions. The first is to improve our model's ability on learning effective information. The error analyses on global and local interactions have shown that our model still faces some problems in producing proper representations. Exploiting more specific strategies to distinguish topic, discourse and other types of words in global interactions can be one choice; or adding matrix operations to deal with sparse local interactions. Another direction lays in providing more interesting recommendation results. Our current model can only predict whether a user will reply to a given conversation. However, sometimes we may need more precise predictions. For example, which turn is the user tends to reply to? Or even what content the user will reply to? However, some users would like to participate in different kinds of conversations, which means that providing diverse recommendations is also crucial. All these functions may exceed the capacity of our current model, and need more effort to explore.

ACKNOWLEDGMENTS

We thank the reviewers for the insightful suggestions on various aspects of this work.

REFERENCES

- [1] Lada A. Adamic and Natalie S. Glance. 2005. The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD'05)*, Jafar Adibi, Marko Grobelnik, Dunja Mladenic, and Patrick Pantel (Eds.). ACM, 36–43.
- [2] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM'08)*, Marc Najork, Andrei Z. Broder, and Soumen Chakrabarti (Eds.). ACM, 183–194.

- [3] Noor Aldeen Alawad, Aris Anagnostopoulos, Stefano Leonardi, Ida Mele, and Fabrizio Silvestri. 2016. Network-aware recommendations of novel tweets. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 913–916.
- [4] David Alvarez-Melis and Martin Saveski. 2016. Topic modeling in Twitter: Aggregating tweets by conversations. In *Proceedings of the 10th International Conference on Web and Social Media*. AAAI Press, 519–522.
- [5] Marcelo Gabriel Armentano, Daniela Godoy, and Analía A. Amandi. 2013. Follower recommendation based on text analysis of micro-blogging activity. *Inf. Syst.* 38, 8 (2013), 1116–1127.
- [6] Yoav Artzi, Patrick Pantel, and Michael Gamon. 2012. Predicting responses to microblog posts. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 602–606.
- [7] Lars Backstrom, Jon M. Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. 2013. Characterizing and curating conversation threads: Expansion, focus, volume, re-entry. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining '13*, Stefano Leonardi, Alessandro Panconesi, Paolo Ferragina, and Aristides Gionis (Eds.). ACM, 13–22.
- [8] Eytan Bakshy, Solomon Messing, and Lada A. Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348, 6239 (2015), 1130–1132.
- [9] David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on Twitter. In *Proceedings of the 9th International Conference on Web and Social Media (ICWSM'15)*. AAAI Press, 574–577.
- [10] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. 2019. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.* 52, 1 (2019), 1–37.
- [11] 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'18)*. Association for Computational Linguistics, 273–283.
- [12] Bin Bi and Junghoo Cho. 2016. Modeling a retweet network via an adaptive bayesian approach. In *Proceedings of the 25th International Conference on World Wide Web*. 459–469.
- [13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (Jan. 2003), 993–1022.
- [14] John S. Breese, David Heckerman, and Carl Kadie. 2013. Empirical analysis of predictive algorithms for collaborative filtering. arXiv:1301.7363. Retrieved from <https://arxiv.org/abs/1301.7363>.
- [15] Ceren Budak and Rakesh Agrawal. 2013. On participation in group chats on Twitter. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 165–176.
- [16] Jonathan P. Chang and Cristian Danescu-Niculescu-Mizil. 2019. Trouble on the horizon: Forecasting the derailment of online conversations as they develop. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 4742–4753.
- [17] 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'11)*. 217–226.
- [18] 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*. ACM, 661–670.
- [19] Tao Chen, Xiangnan He, and Min-Yen Kan. 2016. Context-aware image tweet modelling and recommendation. In *Proceedings of the 24th ACM International Conference on Multimedia*. 1018–1027.
- [20] Wen-Yen Chen, Dong Zhang, and Edward Y. Chang. 2008. Combinational collaborative filtering for personalized community recommendation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 115–123.
- [21] Hao Cheng, Hao Fang, and Mari Ostendorf. 2017. A factored neural network model for characterizing online discussions in vector space. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2296–2306.
- [22] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. 2014. Can cascades be predicted? In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 925–936.
- [23] 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 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'17)*. ACM, 1217–1230.
- [24] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2015. Antisocial behavior in online discussion communities. In *Proceedings of the 9th International Conference on Web and Social Media (ICWSM'15)*. 61–70.
- [25] Hongjian Dou, Wayne Xin Zhao, Yuanpei Zhao, Daxiang Dong, Ji-Rong Wen, and Edward Y. Chang. 2018. Predicting the popularity of online content with knowledge-enhanced neural networks. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD'18)*.

- [26] 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*. Association for Computational Linguistics, 295–303.
- [27] Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander F. Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 154–164.
- [28] Frédéric Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*. 593–596.
- [29] Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'16)*. 2782–2788.
- [30] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 6645–6649.
- [31] Alex Graves and Jürgen Schmidhuber. 2005. Framework phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* 18, 5-6 (2005), 602–610.
- [32] Kishaloy Halder, Min-Yen Kan, and Kazunari Sugiyama. 2019. Predicting helpful posts in open-ended discussion forums: A neural architecture. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19), Volume 1 (Long and Short Papers)*. 3148–3157.
- [33] 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'17)*. 173–182.
- [34] Jack Hessel and Lillian Lee. 2019. Something's brewing! early prediction of controversy-causing posts from discussion features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19), Volume 1 (Long and Short Papers)*. 1648–1659.
- [35] Liangjie Hong and Brian D. Davison. 2010. Empirical study of topic modeling in Twitter. In *Proceedings of the 1st Workshop on Social Media Analytics*. ACM, 80–88.
- [36] 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 6th ACM International Conference on Web Search and Data Mining*. ACM, 557–566.
- [37] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 263–272.
- [38] Haoran Huang, Qi Zhang, Xuanjing Huang, Haoran Huang, Qi Zhang, and Xuanjing Huang. 2017. Mention recommendation for Twitter with end-to-end memory network. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'17)*. 1872–1878.
- [39] Haoran Huang, Qi Zhang, Jindou Wu, and Xuanjing Huang. 2017. Predicting which topics you will join in the future on social media. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 733–742.
- [40] 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 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1145–1154.
- [41] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 133–142.
- [42] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1746–1751.
- [43] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from <https://arxiv.org/abs/1412.6980>.
- [44] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv:1312.6114. Retrieved from <https://arxiv.org/abs/1312.6114>.
- [45] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*.
- [46] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [47] Sumeet Kumar and Kathleen M. Carley. 2019. Tree LSTMs with convolution units to predict stance and rumor veracity in social media conversations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19), Volume 1: Long Papers*. 5047–5058.

- [48] 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'16)*, Volume 1: Long Papers. 2114–2123.
- [49] Jing Li, Yan Song, Zhongyu Wei, and Kam-Fai Wong. 2018. A joint model of conversational discourse and latent topics on microblogs. *Computat. Ling.* 44, 4 (2018), 719–754.
- [50] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* 7, 1 (2003), 76–80.
- [51] 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*.
- [52] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, 73–105.
- [53] Annie Louis and Shay B. Cohen. 2015. Conversation trees: A grammar model for topic structure in forums. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1543–1553.
- [54] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP'17) Volume 1: Long Papers*. 605–614.
- [55] Yi Luan, Yangfeng Ji, and Mari Ostendorf. 2016. LSTM based conversation models. arXiv: Retrieved from <https://arxiv.org/abs/1603.09457>.
- [56] Renfeng Ma, Xipeng Qiu, Qi Zhang, Xiangkun Hu, Yu-Gang Jiang, and Xuanjing Huang. 2019. Co-attention memory network for multimodal microblog's hashtag recommendation. *IEEE Trans. Knowl. Data Eng.* (2019).
- [57] Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 1506–1515.
- [58] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 165–172.
- [59] Alexey N. Medvedev, Jean-Charles Delvenne, and Renaud Lambiotte. 2019. Modelling structure and predicting dynamics of discussion threads in online boards. *J. Complex Netw.* 7, 1 (2019), 67–82.
- [60] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. 2013. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 889–892.
- [61] Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*. 2410–2419.
- [62] Matthew Michelson and Sofus A. Macskassy. 2010. Discovering users' topics of interest on Twitter: A first look. In *Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data*. 73–80.
- [63] Courtney Napoles, Aasish Pappu, and Joel R. Tetreault. 2017. Automatically identifying good conversations online (yes, they do exist!). In *Proceedings of the 11th International Conference on Web and Social Media (ICWSM'17)*. 628–631.
- [64] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan M. Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Proceedings of the 8th IEEE International Conference on Data Mining*. 502–511.
- [65] Ye Pan, Feng Cong, Kailong Chen, and Yong Yu. 2013. Diffusion-aware personalized social update recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 69–76.
- [66] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.
- [67] Kechen Qin, Lu Wang, and Joseph Kim. 2017. Joint modeling of content and discourse relations in dialogues. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, Volume 1: Long Papers, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 974–984.
- [68] Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies*. 172–180.
- [69] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*. 399–408.
- [70] Sara Rosenthal and Kathleen McKeown. 2015. I couldn't agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 168.

- [71] Matthew Rowe and Harith Alani. 2014. Mining and comparing engagement dynamics across multiple social media platforms. In *Proceedings of the ACM Web Science Conference (WebSci'14)*. 229–238.
- [72] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the Machine Learning, Proceedings of the 25th International Conference (ICML'08)*, William W. Cohen, Andrew McCallum, and Sam T. Roweis (Eds.). ACM, 880–887.
- [73] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.
- [74] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2000. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC'00)*. 158–167.
- [75] J. Ben Schafer, Dan Frankowski, Jonathan L. Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The Adaptive Web, Methods and Strategies of Web Personalization*. 291–324.
- [76] Mike Schuster and Kuldeep K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Sign. Process.* 45, 11 (1997), 2673–2681.
- [77] 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'18), Volume 1: Long Papers*. 1616–1626.
- [78] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2006. Collaborative filtering for multi-class data using belief nets algorithms. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. IEEE, 497–504.
- [79] Bongwon Suh, Lichan Hong, Peter Pirollo, and Ed. H. Chi. 2010. Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In *Proceedings of the IEEE Second International Conference on Social Computing*. IEEE, 177–184.
- [80] Koji Tanaka, Junya Takayama, and Yuki Arase. 2019. Dialogue-act prediction of future responses based on conversation history. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19), Volume 2: Student Research Workshop*. 197–202.
- [81] Thomas Tran and Robin Cohen. 2000. Hybrid recommender systems for electronic commerce. In *Proceedings of the Knowledge-Based Electronic Markets, Papers from the AAAI Workshop*, Technical Report WS-00-04. AAAI Press, Vol. 40.
- [82] Marco Del Tredici, Diego Marcheggiani, Sabine Schulte im Walde, and Raquel Fernández. 2019. You shall know a user by the company it keeps: Dynamic representations for social media users in NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 4706–4716.
- [83] Lyle H. Ungar and Dean P. Foster. 1998. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*, Vol. 1. Menlo Park, CA, 114–129.
- [84] Robin Van Meteren and Maarten Van Someren. 2000. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, Vol. 30. 47–56.
- [85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008.
- [86] Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. 2014. Collaborative personalized Twitter search with topic-language models. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 53–62.
- [87] Beidou Wang, Can Wang, Jiajun Bu, Chun Chen, Wei Vivian Zhang, Deng Cai, and Xiaofei He. 2013. Whom to mention: Expand the diffusion of tweets by@ recommendation on micro-blogging systems. In *Proceedings of the 22nd International Conference on World Wide Web*. 1331–1340.
- [88] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 448–456.
- [89] Lingzhi Wang, Jing Li, Kingshan Zeng, Haisong Zhang, and Kam-Fai Wong. 2020. Continuity of topic, interaction, and query: Learning to quote in online conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 6640–6650.
- [90] Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019. Topic-aware neural keyphrase generation for social media language. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19), Volume 1: Long Papers*. 2516–2526.
- [91] Zhibo Wang, Jilong Liao, Qing Cao, Hairong Qi, and Zhi Wang. 2014. Friendbook: A semantic-based friend recommendation system for social networks. *IEEE Trans. Mobile Comput.* 14, 3 (2014), 538–551.

- [92] Penghui Wei, Nan Xu, and Wenji Mao. 2019. Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 4786–4797.
- [93] Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. 2019. Hierarchical user and item representation with three-tier attention for recommendation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19), Volume 1 (Long and Short Papers)*. 1818–1826.
- [94] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. 2011. Who says what to whom on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*. ACM, 705–714.
- [95] 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*. Association for Computational Linguistics, 516–525.
- [96] Yang Yu, Xiaojun Wan, and Xinjie Zhou. 2016. User embedding for scholarly microblog recommendation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 2. 449–453.
- [97] Victoria Zayats and Mari Ostendorf. 2018. Conversation modeling on reddit using a graph-structured LSTM. *Trans. Assoc. Comput. Linguist.* 6 (2018), 121–132.
- [98] Jichuan Zeng, Jing Li, Yulan He, Cuiyun Gao, Michael R. Lyu, and Irwin King. 2019. What you say and how you say it: Joint modeling of topics and discourse in microblog conversations. *Trans. Assoc. Comput. Linguist.* 7 (2019), 267–281.
- [99] 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'18), Volume 1: Long Papers*. 375–385.
- [100] 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'19), Volume 1: Long Papers*. 2809–2818.
- [101] Xingshan Zeng, Jing Li, Lu Wang, and Kam-Fai Wong. 2019. Neural conversation recommendation with online interaction modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 4625–4635.
- [102] Justine Zhang, Jonathan P. Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18), Volume 1: Long Papers*. 1350–1361.
- [103] Justine Zhang, Cristian Danescu-Niculescu-Mizil, Christina Sauper, and Sean J. Taylor. 2018. Characterizing online public discussions through patterns of participant interactions. *Proc. ACM Hum. Comput. Interact.* 2 (2018), 198:1–198:27.
- [104] Qi Zhang, Yeyun Gong, Ya Guo, and Xuanjing Huang. 2015. Retweet behavior prediction using hierarchical dirichlet process. In *Proceedings of the AAAI Annual Conference on Artificial Intelligence (AAAI'05)*. 403–409.
- [105] Wei Zhang, Jianyong Wang, and Wei Feng. 2013. Combining latent factor model with location features for event-based group recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 910–918.
- [106] Yingyi Zhang, Jing Li, Yan Song, and Chengzhi Zhang. 2018. Encoding conversation context for neural keyphrase extraction from microblog posts. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18), Volume 1 (Long Papers)*. 1676–1686.
- [107] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing Twitter and traditional media using topic models. In *Proceedings of the European Conference on Information Retrieval*. Springer, 338–349.

Received October 2020; revised April 2021; accepted July 2021