Dynamic Online Conversation Recommendation

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

Abstract

Trending topics in social media content evolve over time, and it is therefore crucial to understand social media users and their interpersonal communications in a dynamic manner. In this research we study *dynamic online conversation recommendation*, to help users engage in conversations that satisfy their evolving interests. Different from works in conversation recommendation which assume static user interests, our model captures the temporal aspects of user interests. Moreover, our model can cater for cold start problem where conversations are new and unseen in training. We propose a neural architecture to analyze changes of user interactions and interests over time, whose result is used to predict which discussions the users are likely to enter. We conduct experiments on large-scale collections of Reddit conversations. Results on three subreddits show that our model significantly outperforms state-of-the-art models based on static assumption of user interests. We further evaluate performance in cold start, and observe consistently better performance by our model when considering various degrees of sparsity of user's chatting history and conversation contexts. Lastly, our analysis also confirms the change of user interests. This further justify the advantage and efficacy of our model.

1 Introduction

Online social media platforms are popular outlets for individuals to exchange viewpoints and discuss topics they are interested in. However, the huge volume of online conversations produced daily hinders people's capability of finding the information they are interested in. As a result, there is pressing demand for developing a conversation recommendation engine that tracks ongoing conversations and recommends suitable ones to users.

Viewing the deluge of information streaming through social media, it is not hard to envision that

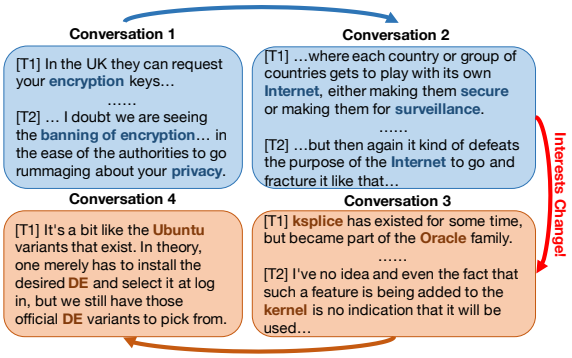


Figure 1: Four chatting snippets posted by the same user U on Reddit. Arrows linking conversation 1 to 4 follow the chronological order. U's interests shifted from *Internet security* (conversations 1 and 2) to *operation system* (conversation 3 and 4).

users' tastes, stances, and behaviors evolve over time (Wu et al., 2017). Nonetheless, existing work on recommending conversations (Chen et al., 2011; Zeng et al., 2018, 2019b) assume users' discussion preferences do not change over time. Moreover, the common practice of recommendation is via collaborative filtering (CF), which relies on rich user interaction history for model training (Zeng et al., 2018, 2019b). When a conversation is entirely absent from training data, the model performance is inevitably compromised. This phenomenon is referred to as **conversation cold start**. As a result, existing methods which ignore the time-evolving user interests is insurmountable to tackle a common problem in practice, i.e., to predict future conversations created after the model is trained.

To overcome this predicament, we explore dynamic conversation recommendation, which can model the change of user interests over time (henceforth **user interest dynamics**). To illustrate such change, Figure 1 shows multiple conversation turns posted by user U in four Reddit discussion snippets: C_1 to C_4 in the chronological order. As can

be seen, U used to like discussing *Internet security*, indicated by "encryption", "privacy", and "surveillance" in C_1 and C_2. After a period of time, U's interests changed to a different topic, *operating system*, as "ksplice", "oracle", and "Ubuntu" were later mentioned in C_3 and C_4.

We design the model to capture user interests from both what they said in the past, and how they interacted with each other in the conversation structure. We first capture time-variant representations from user chatting history, where we assume user interests may change over time and therefore apply a gated recurrent unit (GRU) (Cho et al., 2014) to model time dependency. User interactions in the conversation context are then explored with both bidirectional gated recurrent unit (Bi-GRU) (Cho et al., 2014) for conversation turns' chronological order and graph convolutional networks (GCN) (Marcheggiani and Titov, 2017) for in-reply-to relations. Both representations are learned to encode how participants formed the conversation structure, including what they said and whom they replied to. Next, we propose a user-aware attention to convey the user interest dynamics, which is further put over an interaction-encoded conversation to measure whether its ongoing contexts fit a user's current interests. Finally, we predict how likely a user will engage in a conversation, as a result of recommendation. To the best of our knowledge, *we are the first to study dynamic online conversation recommendation and to explore the effects of user interests change over time learned from both chatting content and interaction behavior*. For this reason, we are capable of recommending future conversations based on users' interests at the time.

For experiments[1], we collect Reddit conversations from three subreddits — "*technology*", "*todayilearned*", and "*funny*", each exhibiting different data statistics, discussion topics, and language styles. An absolute date is used to separate training data (before the date) from test and validation data (after the date). In this way, most conversations in the test and validation parts are new conversations that have not been counted before. This presents a more realistic setup than previous studies (Zeng et al., 2018, 2019b), which let training data contain partial context for any conversations to allow the possibility of predicting users' future engagement for recommendation.

Experimental results in main comparisons show that our model significantly outperforms all previous methods that ignore the change of user interests or interactions within contexts. For example, we achieve 0.375 MAP in discussions of "*technology*", compared with 0.222 yielded by our previous state-of-the-art model (Zeng et al., 2019b). Further study shows that we consistently perform better both in conversation cold start and with varying degrees of sparsity of user history and conversation contexts. Lastly, to provide more insights into user interest dynamics, we inspect our model outputs and find that users indeed tend to engage in different types of conversations at different times, confirming the usefulness of tracking user preferences in real-time for conversation recommendation.

2 Related Work

User Response Prediction. This work is in line with user response prediction, such as message popularity forecast with handcrafted response features (Artzi et al., 2012; Backstrom et al., 2013) and conversation trajectory with user interaction structures (Cheng et al., 2017b; Jiao et al., 2018; Zeng et al., 2019a). These works predict responses from general public, while we work on personalized recommendation and focus on user interest modeling. For recommendation, there are extensive efforts on post-level recommendation (Chen et al., 2012; Yan et al., 2012) and conversation-level (Chen et al., 2011; Zeng et al., 2018, 2019b). In contrast with them which assume static user interests, we capture how user interests change over time and take advantage of the recent advancement of dynamic product recommendation (Wu et al., 2017; Beutel et al., 2018). To recommend conversations, we aim to learn user interest dynamics from chatting content and interaction behavior, which have never been explored in previous research.

Conversation Structure Modeling. Our work is also related to previous work to understand how participants interact with each other in conversation structure. Earlier efforts focus on discovering word statistic patterns via probabilistic graphical models (Ritter et al., 2010; Louis and Cohen, 2015), which are unable to capture deep semantics embedded in complex interactions. Recent research points out the effectiveness to understand conversation structure from temporal dynamics (Cheng et al., 2017a; Jiao et al., 2018) and replying struc-

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

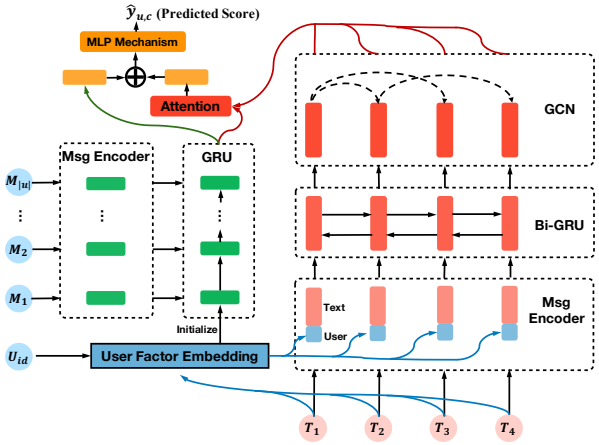


Figure 2: Overall structure of our model. The left module is to model user interest dynamics, whose results together with conversation representations derived from the right part are used for producing final prediction. Predicted score $\hat{y}_{u,c}$ indicates how likely u will engage in c. "Msg Encoder" mainly contains two layers: word embedding layer and CNN modeling layer.

ture (Miura et al., 2018; Zayats and Ostendorf, 2018; Zeng et al., 2019b). The two factors are coupled in our interaction modeling and their joint effects for dynamic conversation recommendation, ignored by prior work, will be extensively studied here.

3 Our Dynamic Conversation Recommendation Model

This section describes our dynamic conversation recommendation model, whose overall structure is shown in Figure 2. In the following, we will first introduce how we model the user interest dynamics with their chatting history in Section 3.1, followed by the description of conversation modeling in Section 3.2. Afterwards, Section 3.3 will present how we produce final recommendation outputs. Objective function and learning procedures will be finally presented in Section 3.4.

3.1 User Interest Dynamic Modeling

Given a sequence of chronologically ordered historical messages $\langle m_1, m_2, \cdots, m_{|u|}\rangle$ of a user u ($|u|$ is the message number of u), a message therein corresponds to a word sequence $\mathbf{w}_m$. Our goal is to capture the temporal patterns from the sequence of user chatting messages and then produce the user interest representation. We employ two-level modeling — message level and user level.

Message-level Modeling. We model message-level representation from its word sequence. Specifically, given u's historical message m, we first use a pre-trained word embedding layer to map each word into a vector space, and then employ a Convolutional Neural Network (CNN) (Kim, 2014) encoder to model word occurrence with their neighbors. Afterwards, we output representation $\boldsymbol{z}_m$ to reflect m's content.

User-level Modeling. As shown in Wu et al. (2017), some user interests may change rapidly and some may last for a long time. For the latter, we adopt a user embedding layer $I^{UF}(\cdot)$ to capture the time-invariant interest factor and define u's factor as $\boldsymbol{r}_u^{UF}$.

For the time-variant interests, we are inspired by previous work (Beutel et al., 2018) and employ a GRU (Cho et al., 2014) encoder to capture how user interests change based on sequential chatting messages. For each time state t, we update user's current interests $\boldsymbol{h}_{u,t}^U$ conditioned on the previous interests $\boldsymbol{h}_{u,t-1}^U$ and the current behavior $\boldsymbol{z}_{m_t}$ (derived from the aforementioned message-level modeling, reflecting m's content):

$$\boldsymbol{h}_{u,t}^U = GRU(\boldsymbol{h}_{u,t-1}^U, \boldsymbol{z}_{m_t}) \quad (1)$$

Further, to leverage time-invariant features in the modeling of user interest dynamics, we initialize GRU's hidden states based on the learned user factor $\boldsymbol{r}_u^{UF}$ following linear transformation: $\boldsymbol{h}_{u,0}^U = W^U \boldsymbol{r}_u^{UF} + b^U$. And the last GRU states, i.e., $\boldsymbol{r}_u^U = \boldsymbol{h}_{u,t_{|u|}}^U$, conveying the latest view of user interest dynamics, will be later used in conversation modeling and recommendation prediction.

3.2 User-aware Conversation Modeling

Here we introduce how we encode a conversation in aware of user interests. Each conversation c is formed with a sequence of chronologically ordered turns $\langle t_1, t_2, ..., t_{|c|}\rangle$ ($|c|$ is the turn number of c). A turn t therein is in form of a word sequence $\mathbf{w}_t$, its author's ID u_t, and the turn it replies to for later exploiting in-reply-to structure.

To learn c's representation, we encode both word occurrence in each turn (via turn-level modeling) and interactions between conversation turns (via conversation-level modeling). Afterwards, to identify turns that match target user's interests, we propose a user-aware attention over turns.

Turn-level Modeling. For each turn $t \in c$, similar to message-level modeling in Section 3.1, we use a CNN encoder over pre-trained word embeddings to capture content representation, $\boldsymbol{z}_t$. Further, $\boldsymbol{z}_t$ is concatenated with author u_t's user embedding $\boldsymbol{r}_{u_t}^{UF}$ (see Section 3.1) to yield turn-level representation $\boldsymbol{r}_t^T$, conveying both what is said and who says that. Based on the turn-level representations, we then learn turn interactions.

Conversation-level Modeling. To explore turn interactions, we exploit turn's chronological order and replying structure, both useful in conversation modeling (Zeng et al., 2019b).

Chronological Order. We employ a Bi-GRU (Cho et al., 2014) to capture how a turn interacts with the turns posted right before and after it, whose hidden states are updated as followings:

$$\overrightarrow{\boldsymbol{h}_{c,t}^{GRU}} = \overrightarrow{GRU}(\boldsymbol{h}_{c,t-1}^{GRU}, \boldsymbol{r}_t^T) \tag{2}$$

$$\overleftarrow{\boldsymbol{h}_{c,t}^{GRU}} = \overleftarrow{GRU}(\boldsymbol{h}_{c,t+1}^{GRU}, \boldsymbol{r}_t^T) \tag{3}$$

We then concatenate the forward and backward hidden states to produce chronology-encoded turn representations: $\boldsymbol{h}_{c,t}^{GRU} = [\overrightarrow{\boldsymbol{h}_{c,t}^{GRU}}; \overleftarrow{\boldsymbol{h}_{c,t}^{GRU}}]$.

Replying Structure. To further encode who-replies-to-whom in conversation structure, we put a Graph Convolutional Network (GCN) (Marcheggiani and Titov, 2017) over the chronology-encoded turn representations (learned by Bi-GRU see above). Graph encoder is empirically better than sequential ones because replying relations usually exhibit tree structure (a post may lead to multiple replies). Concretely, we first build a directed graph for a conversation via adding edges from a turn to its replies. We then define turn interactions therein in three directions: predecessors to successors (Pre), successors to predecessors (Suc), and self interactions ($Self$). Next, we update a turn's hidden state with the formula below:

$$\begin{aligned}\boldsymbol{h}_{c,t}^{GCN} = &\sum_{i \in Pre(t)} g_{i,t}(W^{Pre}\boldsymbol{h}_{c,i}^{GRU} + b^{Pre}) + \\ &\sum_{j \in Suc(t)} g_{j,t}(W^{Suc}\boldsymbol{h}_{c,j}^{GRU} + b^{Suc}) + \\ &g_{t,t}(W^{Self}\boldsymbol{h}_{c,t}^{GRU} + b^{Self})\end{aligned} \tag{4}$$

$Pre(t)$ and $Suc(t)$ represent turn t's predecessors and successors in replying graph; $g_{i,j}$ is a scalar gate controlling weights of turn interactions:

$$g_{i,j} = \sigma(W^{Dir(i,j)}\boldsymbol{h}_{c,i}^{GRU} + b^{Dir(i,j)}) \tag{5}$$

where $Dir(i,j)$ indicates the type of i-j direction (Pre, Suc, or $Self$).

The process described above can be viewed as one GCN layer. Multiple layers can be stacked, with a ReLU (Rectified Linear Unit) activated function to connect two succinct layers. It enables the networks to explore deeper interaction effects.

User-aware Attention. To identify conversation turns that better match target user's interests, we design a user-aware attention mechanism over interaction-encoded turns. The attention weights are defined to reflect the similarity between a conversation turn's representation $\boldsymbol{h}_{c,i}^{GCN}$ and the target user's latest interests $\boldsymbol{r}_u^U$ (see Section 3.1):

$$a_i = softmax(\boldsymbol{r}_u^U \cdot \boldsymbol{h}_{c,i}^{GCN}) \tag{6}$$

Finally, we compute the attentive sum of all turns and obtain the conversation representations conveying both interactions and user interests:

$$\boldsymbol{r}_c^C = \sum_i a_i \boldsymbol{h}_{c,i}^{GCN} \tag{7}$$

3.3 Recommendation Prediction

To predict whether a user u willengage in conversation c, we compute how u's interest dynamics (carried by $\boldsymbol{r}_u^U$ in Section 3.1) are similar to c's content and interaction styles (reflected by $\boldsymbol{r}_c^C$ in Section 3.2). We adopt a two-way interactions via MLP mechanism (He et al., 2017) to measure the similarity:

$$\boldsymbol{r}_{u,c} = \alpha(W_2^T(\alpha(W_1^T[\boldsymbol{r}_u^U; \boldsymbol{r}_c^C] + b_1)) + b_2) \tag{8}$$

where $\alpha(\cdot)$ is ReLU-activated function.

For recommendation, we predict $\hat{y}_{u,c} \in [0,1]$, which signals how likely u will engage in c. The equation for the final output layer will be:

$$\hat{y}_{u,c} = \sigma(\boldsymbol{v}^T \boldsymbol{r}_{u,c} + b) \tag{9}$$

where σ represents sigmoid activation function.

3.4 Learning Objective

Following Zeng et al. (2019b), we adopt *weighted binary cross-entropy loss* as our objective function, which assigns more weights to positive feedbacks (i.e. u engages in c):

$$\mathcal{L} = -\sum_{(u,c) \in \mathcal{T}} \left[\lambda \cdot y_{u,c} \log(\hat{y}_{u,c}) + (1 - y_{u,c}) \log(1 - \hat{y}_{u,c})\right] \tag{10}$$

	Tech	Learn	Fun
Number of Users	13,927	67,255	112,345
Number of Convs	8,286	42,220	67,908
Number of Turns	43,705	233,213	375,550
Hist Number / User	2.78	3.05	2.94
Turn Number / Conv	5.10	5.34	5.35
User Number / Conv	4.15	4.45	4.79
New User Rate (%)	8.20	8.24	7.81
New Conv Rate (%)	99.64	99.40	99.51

Table 1: Data statistics. "Conv": conversation; "Hist": historical messages. New user rate is the number of users newly appeared in May's data (for test) divided by number of May's users. New conversation rate is similar.

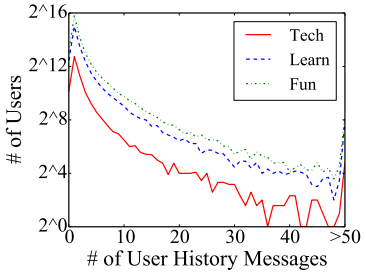
where $\mathcal{T}$ is the training set, $y_{u,c}$ denotes the binary ground-truth label, and λ ($\lambda > 1$) is a hyper-parameter to trade off the weights of positive and negative instances. We weigh more on positive feedbacks because they are more reliable, while the negative ones sometimes cannot reflect user's interests, owing to many unpredictable issues (e.g., users' busy time). For the same reason, we adopt the negative sampling strategy (He et al., 2017) in training, which also speeds up the training process.

4 Experimental Setup

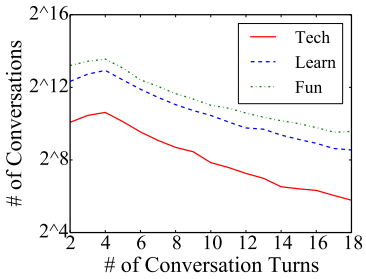
Datasets. For experiments, we collect online conversations from Reddit, a popular online platform. To build our datasets, we first downloaded a large corpus publicly available on Reddit[2], which consists of posts and comments created since early 2006. Then, we gathered data posted from January to May 2015 on three subreddits reflecting discussion topics on "*technology*" (**Tech**), "*todayilearned*" (**Learn**), and "funny" (**Fun**). We chose these three subreddits as they were popular subreddits with different discussed topics and language styles. For each subreddit, posts and comments were connected with in-reply-to relations (indicated by comments' "*parent_id*" field) to form conversations. Finally, we removed conversations with only one turn and produced three conversation datasets of different topics.

In model training and evaluation, we use conversation turns created from January to April for training. For those posted in May, we randomly select half of them for validation and the other half for test. This reflects a more realistic scenario where the model is trained with past data and applied to future recommendation, as opposed to prior work which assumes all conversations can be split between training and test (Zeng et al., 2018, 2019b).

[2] https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/



(a) User History Dist.



(b) Conversation Turn Dist.

Figure 3: Distribution of users' historical message count (upper) and conversation turn count (lower).

Data Analysis. The dataset statistics are displayed on Table 1. Although differ in size, conversations therein exhibit similar average characteristics, likely because they come from the same platform. Moreover, over 99% of the conversations in test sets are future conversations (i.e. all turns were posted in May), highlighting the challenge of conversation cold start.

We further plot the distributions of message (turn) number in Figure 3 (3(a) for users and 3(b) for conversations). It is seen from Figure 3(a) that a large proportion of users were involved in less than 10 conversation turns, where about 8% (shown in Table 1) of users are absent in the training data. For conversations (Figure 3(b)), their turn numbers follow a power-law distribution. Therefore, for both users and conversations, the sparse interaction history presents additional challenges for recommendation.

In addition, Figure 4 shows distributions of conversation replying structure with 1, 2, and more root-to-leaf paths to characterize users' interaction structure. We find that more than 60% of con-

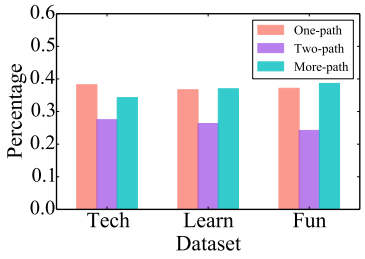


Figure 4: Distributions of conversation structure. "One-path", "Two-path", and "More-path" indicate the conversation has 1, 2, and more root-to-leaf paths.

versations contain two or more paths, illustrating complex who-replies-to-whom interactions in the tree structure (with the original post as the root node and in-reply-to relations as edges). Therefore, graph-structured encoder may be a suitable alternative for capturing rich turn interactions in Reddit conversations.

Preprocessing. For all datasets, we applied open source natural language toolkit (NLTK) (Loper and Bird, 2002) for tokenization. Further, links were replaced by a generic tag "⟨URL⟩" and all number tokens were removed. In the experiments, we maintained a vocabulary with all the remaining tokens (including punctuation and emoticons).

Model Settings. In training, we adopt negative sampling with sampling ratio of 5 (see Section 3.4). We also randomly sample 100 negative instances for each positive one during validation and test, to avoid unbalanced labels.

For parameters, we initialize the word embedding layer with 300-dim Common Crawl version of Glove embedding (Pennington et al., 2014), and the dimension of user factor embedding is set to 20. For the CNN turn encoders, we use filter windows of 2, 3, and 4, each with 100 feature maps. As for the GRU models for both user and conversation modeling, the hidden state size is set to 200 (100 for each direction in Bi-GRU). The same hidden state size is applied to the GCN interaction model. We also set the layer number of GCN (see in Section 3.2) to 1, based on validation results. In training, the batch size is set to 256 and Adam optimizer (Kingma and Ba, 2014) is adopted with an initial learning rate of 0.001. As for the trade off weight in loss function, we set $\lambda = 100$.

Evaluation. Our evaluation metrics follow the common practice in conversation recommendation (Zeng et al., 2018, 2019b). Mean average precision (MAP), precision at 1 (P@1), and normalized Discounted Cumulative Gain at 5 (nDCG@5) are adopted to measure the ranking list of conversations to be recommended to a user.[3] These metrics all have a value range of 0.0 to 1.0, and greater value indicates better performance.

Comparisons. We first consider two simple baselines: 1) ranking conversations based on POPULARITY, measured by the number of participants. 2) TOPICRANK (Chen et al., 2011): ranking conversations by topic relevance to the target user's historical messages, where topics are learned from both LDA (Blei et al., 2003) and TF-IDF statistics.

We also include previous conversation recommendation models without learning user interest dynamics: 3) CRJTD (Zeng et al., 2018): a CF-based method that jointly models topics and discourse with LDA-style Bayesian models. 4) CRIM (Zeng et al., 2019b): a neural CF framework with GCN-based interaction modeling, which presents state-of-the-art conversation recommendation results in previous work.

In addition, we compare with the following recent models for product recommendation. 5) RRN (Wu et al., 2017): exploiting RNN model to capture user interest dynamics only with user interaction history (without modeling turn content). 6) LC-RNN (latent cross-RNN) (Beutel et al., 2018): RNN-based user interest dynamic modeling with turn-level representations, with participant interactions in the conversation structure ignored.

5 Experimental Results

We first report the main comparison results in Section 5.1, and then discuss the effects of sparsity and cold start in Section 5.2. Lastly, in Section 5.3, we probe into our model outputs to provide more insights into user interest dynamics.

5.1 Main Comparison Results

Table 2 shows the comparison results on all three datasets. Our model achieves the highest scores, outperforming all comparison models by a large margin. It suggests that dynamic user interests learned from both content and interactions provide clearly useful signals on which conversations a user is likely to engage in. Below describes more detailed observations.

[3] We also experiment with nDCG@10, and same trend holds.

Models	Tech			Learn			Fun		
	MAP	P@1	nDCG	MAP	P@1	nDCG	MAP	P@1	nDCG
Simple Baselines									
POPULARITY	0.055	0.012	0.031	0.057	0.012	0.033	0.058	0.011	0.033
TOPICRANK (Chen et al., 2011)	0.087	0.037	0.071	0.071	0.031	0.050	0.065	0.024	0.042
Unchanged Interests									
CRJTD (Zeng et al., 2018)	0.193	0.173	0.184	0.158	0.135	0.150	0.113	0.085	0.101
CRIM *(SOTA)* (Zeng et al., 2019b)	0.222	0.180	0.187	0.204	0.151	0.194	0.162	0.114	0.150
Dynamic Interests									
RRN (Wu et al., 2017)	0.190	0.210	0.199	0.221	0.270	0.238	0.190	0.227	0.201
LC-RNN (Beutel et al., 2018)	0.212	0.222	0.234	0.222	0.294	0.240	0.198	0.255	0.211
OURS	**0.375**	**0.391**	**0.369**	**0.347**	**0.368**	**0.344**	**0.283**	**0.294**	**0.274**

Table 2: Results of our main experiments (averaged over users). "nDCG" stands for "nDCG@5". CRIM is from our prior work which obtained previous state-of-the-art. The best result for each column is in **boldface**. Our model significantly outperforms all comparisons ($p < 0.01$, paired t-test).

The two baselines yield much worse results than others. This shows the challenging nature of conversation recommendation, and the limitation of simply using popularity or topic similarity. TOPICRANK performs slightly better than POPULARITY, indicating that individuals are more inclined to engage in conversations they like (reflected by topic relevance), rather than popular discussions with many participants.

Our model outperforms CRJTD and CRIM (state-of-the-art model), which both assume fixed user interests, showing the usefulness of exploring user's evolving interests over time. We also find that CRIM produces better results than CRJTD, likely because the former additionally captures user interactions among each other.

For recommendation models that consider user interest dynamics, all models perform better than CRIM and CRJTD, which are both based on the CF architecture. This reveals CF's limitation in dealing with cold start, which is a common phenomenon when recommending a large number of future conversations (see Table 1). Nevertheless, we see that our model performs much better than RRN and LC-RNN, indicating that both content and interaction features contribute to capturing user interests and how they change over time.

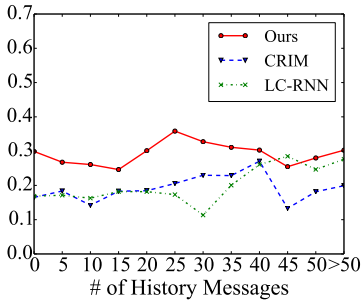
5.2 History Sparsity and Cold Start

Similar to previous work in product recomendation (Sarwar et al., 2000), conversation recommendation models are also susceptible to the problems of history sparsity and cold start. We compare with LC-RNN (the best comparison model in Table 2) and CRIM (state-of-the-art model in conversation recommendation), and show in Figure 5 the MAP scores on Tech dataset with varying degrees of sparsity.[4] Our model is shown to be consistently better in face of sparsity, including varying numbers of messages in user history, as well as varying numbers of available turns in conversation contexts. More detailed discussions are presented below.

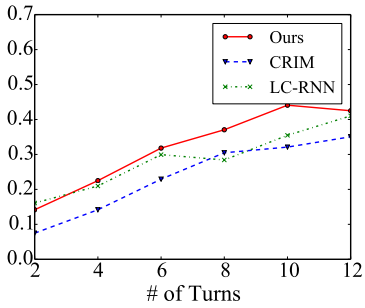
Varying Messages in User History. Refer to in Figure 5(a), all models produce non-monotonic performance curves, peaking at certain points (e.g. 25 historical messages for our model). This reveals the issue of user history sparsity, and difficulty in coping with excessive historical information. More importantly, it is observed that our model already outperformed LC-RNN and CRIM when the number of history message is 0. This may be attributed to our better modeling on conversation interaction structure.

Varying Turns in Conversation Context. For conversations, Figure 5(b) shows the MAP scores with varying turn numbers available in contexts. All three models produce upward-trending curves, which is expected since more features can be learned from richer contexts, thus leading to better prediction. Our model and CRIM perform worse than LC-RNN when available turn number is small (less than 4). This is because graph-structured networks need minimum amount of interaction infor-

[4] Similar trends are observed on all datasets and hence only the results on Tech are displayed.



(a) User History



(b) Conversation Context

Figure 5: MAP scores on Tech dataset with varying degrees of sparsity in user chatting history (upper) or conversation context (lower). Our model performs consistently better.

mation for effective modeling of the conversation structures.

Conversation Cold Start. To understand how models perform exactly in conversation cold start, we separate the test set into future conversations (newly created in testing and unseen in training data) and existing ones (with context partially in the training data). We then compute the results averaging over conversations. The resultant MAP scores are reported in Table 3. Our model outperforms the other two models by a large margin in recommending future conversations, thanks to the more accurate user interests that are learned from dynamic patterns of content and interactions. CRIM performs much better for existing conversations, by making use of rich user interaction history based on CF architecture. Our model abandons CF framework but still produce competitive performance, as we compute more accurate user-aware representations.

5.3 More Analyses on Our Model

The aforementioned results have shown the efficacy and advantage of our model. In this section, we provide more insights into different factors behind the model, in order to obtain a better understanding of its performance.

Models	Future Convs			Existing Convs		
	Tech	Learn	Fun	Tech	Learn	Fun
CRIM	0.208	0.165	0.142	**0.684**	0.731	0.455
LC-RNN	0.214	0.220	0.197	0.129	0.587	0.318
OURS	**0.384**	**0.356**	**0.305**	0.590	**0.749**	**0.458**

Table 3: MAP scores to predict future and existing conversations (averaged over conversations). Our model performs the best in conversation cold start.

Training with More History. We have shown the usefulness of capturing user interest dynamics with historical messages. A natural question is whether the model needs more history to perform better. Figure 6 shows our MAP scores trained on history data in the last x months ($x = 1, 2, 3, 4$), and the three datasets exhibit diverse characteristics in user interest dynamics. Only Tech exhibits an increasing trend. This is probably because earlier history enables learning of long-term dynamics and technology change usually happens in a time span that is longer than 1-2 months. On the contrary, topics on Fun and Learn may change more rapidly, making the earlier history more noisy and less helpful for modeling users' current interests.

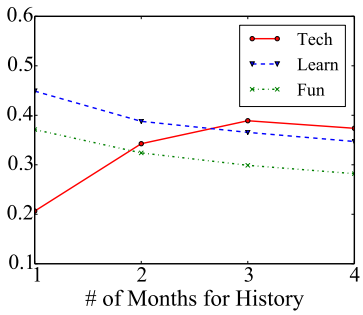


Figure 6: MAP scores of our model with training data in the last x months.

Ablation Study. We then examine the contributions of different components in our model, and display the MAP scores of various ablations in Table 4. We observe that user factor embedding and user-aware attention contribute most to model outputs because they are critical in modeling user interests. Removing Bi-GRU or GCN also has a significant impact on performance, indicating the usefulness of learning user interactions from turn chronology and replying relations.

To further understand the effects of Bi-GRU and

Models	Tech	Learn	Fun
w/o user factor embedding	0.174	0.159	0.122
w/o user-aware attention	0.188	0.183	0.149
w/o Bi-GRU	0.299	0.253	0.206
w/o GCN	0.276	0.307	0.221
Our full model	**0.375**	**0.347**	**0.283**

Table 4: MAP scores with different parts ablated. The best MAP results are highlighted in **bold**.

GCN in user interaction modeling, we compare the MAP scores of our full model and its variants without Bi-GRU or GCN in recommending conversations with 1, 2, or more root-to-leaf paths (as shown in Figure 7). GCN and Bi-GRU clearly demonstrate different capabilities. The former is good at encoding more complex structures (i.e. those with more paths), and the latter excels at sequential conversations. By leveraging the advantages of both, our full model performs the best for conversations of varying structures.

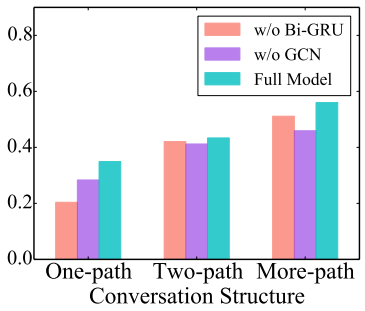


Figure 7: Results of our full model and its variants without Bi-GRU or GCN for recommending conversations in different structures. X-axis: number of root-to-leaf paths. Y-axis: MAP scores.

Case Study. Lastly, we use the example in Figure 1 to analyze what the model has learned for recommendation. Recall that user U's interests shifted from *Internet security*, signaled earlier in C_1 and C_2, to *operation system*, when later chatting in C_3 and C_4. We examine the predicted likelihoods of U engaging in two future conversations: Conversation A and B. Figure 8 shows their contexts—A focuses on *Internet security* and B on *file system*, and U later engaged in B but not A due to the interest shift. In Table 5, we list our model's outputs when fed with earlier history only (C_1 and C_2), later only (C_3 and C_4), and full history, respectively. Not surprisingly, much higher scores are given to A when only the earlier history is given, as it fits well with U's previous preference. Similarly, we correctly predict U to engage in B with much higher confidence in the other two situations as *file system* (B's focus) and *operation system* (U's later interests) are highly related. Given the full history, our model produces more closed scores, showing its efficacy of learning user interest dynamics.

Conversation A
$[T_1]$: Ahhh! This reminds me of when you could *hack* fax machines and *routers* by just whistling in the phone!
$[T_2]$: Hm, that's pretty unrelated, though..
...

Conversation B
$[T_1]$: ...just downloaded *FileZilla* (from SourceForge) last night, and it automatically installed *MacKepper* and...
$[T_2]$: Dude, why? *Filezilla* has a website, you can download it straight from them...
...

Figure 8: Context turns in Conversation (Conv.) A and B. *Blue italic* words indicate A's topic—*Internet security* and *red italic* words in B reflects its focus on *file system*.

U's History Given	Conv. A	Conv. B
Earlier history only (C_1, C_2)	0.733	0.267
Later history only (C_3, C_4)	0.297	0.703
Full history (C_1, C_2, C_3, C_4)	0.421	0.579

Table 5: Predicted likelihoods of U entering Conversations A and B. B is ranked higher than A due to shifted user interests.

6 Conclusion

This paper presents a dynamic conversation recommendation model learned from the change of content and user interactions over time. Experimental results on three new datasets from Reddit show that our model significantly outperforms all comparisons, including previous state of the arts. Further discussion demonstrates the robustness of our model against history sparsity and cold start. We also analyze our model's outputs to get more insights into user interest dynamics.

Acknowledgements

The research described in this paper is partially supported by HK RGC-GRF grant #14204118. Jing Li is partly funded by the Hong Kong Polytechnic University internal fund (1-BE2W). Lu Wang is supported by National Science Foundation through Grant IIS-1813341. We thank the three anonymous reviewers for the insightful suggestions on various aspects of this work.

References

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

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

Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 46–54.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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

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

Hao Cheng, Hao Fang, and Mari Ostendorf. 2017a. 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*, pages 2296–2306.

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

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

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

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

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

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

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*.

Annie Louis and Shay B. Cohen. 2015. Conversation trees: A grammar model for topic structure in forums. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1543–1553. Association for Computational Linguistics.

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

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

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

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

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), Minneapolis, MN, USA, October 17-20, 2000*, pages 158–167.

Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pages 495–503.

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

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

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

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

Xingshan Zeng, Jing Li, Lu Wang, and Kam-Fai Wong. 2019b. Neural conversation recommendation with online interaction modeling. 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)*, pages 4625–4635, Hong Kong, China.