Personalized Chit-Chat Generation for Recommendation Using External Chat Corpora

Changyu Chen
Gaoling School of AI (GSAI)
Renmin University of China
chen.changyu@ruc.edu.cn

Xiting Wang
Microsoft Research Asia
xitwan@microsoft.com

Xiaoyuan Yi
Microsoft Research Asia
xiaoyuanyi@microsoft.com

Fangzhao Wu
Microsoft Research Asia
wufangzhao@microsoft.com

Xing Xie
Microsoft Research Asia
xing.xie@microsoft.com

Rui Yan*
Gaoling School of AI (GSAI)
Renmin University of China
ruiyan@ruc.edu.cn

ABSTRACT

Chit-chat has been shown effective in engaging users in human-computer interaction. We find with a user study that generating appropriate chit-chat for news articles can help expand user interest and increase the probability that a user reads a recommended news article. Based on this observation, we propose a method to generate personalized chit-chat for news recommendation. Different from existing methods for personalized text generation, our method only requires an external chat corpus obtained from an online forum, which can be disconnected from the recommendation dataset from both the user and item (news) perspectives. This is achieved by designing a weak supervision method for estimating users' personalized interest in a chit-chat post by transferring knowledge learned by a news recommendation model. Based on the method for estimating user interest, a reinforcement learning framework is proposed to generate personalized chit-chat. Extensive experiments, including the automatic offline evaluation and user studies, demonstrate the effectiveness of our method[1].

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**; • **Computing methodologies** → **Natural language generation**.

KEYWORDS

personalized text generation; news recommendation; chit-chat; reinforcement learning

ACM Reference Format:
Changyu Chen, Xiting Wang, Xiaoyuan Yi, Fangzhao Wu, Xing Xie, and Rui Yan. 2022. Personalized Chit-Chat Generation for Recommendation Using External Chat Corpora. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539215

*Corresponding author: Rui Yan (ruiyan@ruc.edu.cn)
[1]Our dataset and code are publicly available at https://github.com/chen0changyu/PRET

KDD '22, August 14–18, 2022, Washington, DC, USA.

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539215

1 INTRODUCTION

Recommendation has become one of the most widely adopted techniques for handling information overload. As recommender systems impact the daily lives of users in an increasingly profound way, the community has reached a consensus that merely improving recommendation accuracy is not enough. It is also important that we optimize how recommender systems communicate with users to provide a desirable user experience. To achieve this goal, many pioneering works have been proposed to give explanations by the knowledge graph [20, 41] or generate informative or attractive text to improve recommendation experience, such as review-based textual explanations about why specific items are recommended [3, 4, 16, 31, 40], personalized item summarization [33], and conversations for efficiently collecting user feedback [2, 5, 27, 42]. These methods not only increase the probability that a user adopts a recommended item, but can also help users make a better decision, which significantly increases user trust and satisfaction [3, 40].

While existing works in this line are useful and promising, they typically assume that the type of text to be generated (e.g., reviews or product descriptions) can be found in the recommendation dataset. This prevents them from being widely adopted in scenarios where a desirable textual corpus does not exist in the recommendation dataset. An interesting question is: *is it possible to leverage an external textual corpus to improve user experience in recommendation?* For example, there are large-scale open-domain textual corpora online that are engaging, informative, and related to recommended items, such as chats about news articles, songs, and online products in the online forums. These textual corpora are potentially beneficial to improve user experience in various recommendation scenarios. However, they are currently unexplored.

We aim to bridge the research gap and move towards using external text data for improving recommendation experience. In particular, we explore the usefulness of online chit-chat corpora due to two reasons. First, generating human-like chit-chat has been shown effective in engaging users in human-computer interaction [38]. Second, external chit-chat corpora in online forums, e.g.,

News: Deforestation in Brazil's Amazon at highest level since 2006 | Environment

User 1 keywords: paris, europe, city
Chit-chat: Pretty good to know that in Europe they clear land and build cities only for the most desirable locations close to the nature
User 2 keywords: economic, cash, campaign
Chit-chat: Would the UN or US apply economic sanctions under threat of a bombing campaign to halt the deforestation?
User 3 keywords: taylor, swift, song
Chit-chat: A song of Taylor is about reviving the nature. It is nice, Blondie.

Figure 1: Personalized chit-chat for news recommendation.

Reddit[2], involve many interesting discussions about items such as news articles, songs, and movies. In this paper, we use news recommendation as a guiding example, and show how personalized chit-chat for each news article can be generated. As shown in Fig. 1, the chit-chat is engaging and fits users' personal interests. It not only increases the probability that a user reads the news, but also provides a novel way for expanding users' reading interest and handling information cocoons [13]. For example, the news headline about environment protection is not interesting to the users previously, but by creating a personalized chit-chat that fits their interest (e.g., Europe) and communicating with the users in a casual and engaging way, the users may become more interested in the news article. This may encourage the users to read something more distantly related to their clicked news.

Although improving recommendation experience with chit-chat is promising, it is unclear whether it can achieve the desired result for real users in a news recommender system. Generating appropriate, personalized chit-chat for users of a recommender system by using an external chit-chat corpus is also non-trivial. There is no ground-truth for either training or evaluation, so a traditional supervised learning framework cannot be applied. Moreover, the chit-chat corpus and recommendation dataset are disconnected both from the user and item perspectives: the users in the recommendation dataset cannot be found in the chit-chat corpus, and there is no guarantee that every news article (item) in the recommendation dataset can be linked to a news article in the chit-chat corpus. As a result, it is very difficult to decide how we can transfer knowledge from the chit-chat corpus to the recommendation dataset, or model users and chit-chat in a unified way, which is indispensable for generating personalized chit-chat.

To solve these challenges, we make the following contributions.

First, we conduct a user study to achieve a better understanding of whether external text like chit-chat can help recommendation. Analysis shows that an appropriate chit-chat can not only significantly increase the news click probability (+26%) but also encourage users to expand his/her reading interest (the expected number of clicked news increased by 27.6%).

Second, we propose a weak supervision method for quantifying users' personalized interest in any chit-chat post with no ground-truth labels. The basic idea is to create weak labels by transferring news headlines into their correspondence in the chit-chat domain. A news recommendation model is then fine-tuned with the weak labels to transfer its knowledge about users' personalized interest in headlines to the chit-chat space. The key here is how we can eliminate noises during headline transfer. Inspired by the crowdsourcing methods, we generate multiple possible chit-chat posts for each news headline, and treat each post as a weak label given by a worker (low-quality labeler). The reliability of each worker is measured with a relevance model, which allows us to flexibly adjust the weight of each weak label. Results in the collected benchmark dataset demonstrate the effectiveness of our model.

[2]https://files.pushshift.io/reddit/

	Headline		Chit-chat		Increase
	Mean	Std.	Mean	Std.	
%Click	0.501	0.19	0.634	0.16	26%*
#News	99.37	38.1	126.8	37.02	27.6%*

Table 1: Usefulness of chit-chat in increasing the click rate of news (%Click) and the number of news clicked by a user (#News). Statistical significant increase is marked with *.

Third, we propose a reinforcement learning method to generate Personalized chit-chat for news Recommendation using External Text corpora (PRET). In this framework, a reward function is designed to score a generated chit-chat post based on both the users' personalized interest and non-personalized factors, e.g., relevance with the news and expected number of likes for the chit-chat. A generation policy is then gradually learned to output chit-chat that maximizes the reward. The key here is how to ensure that the generated chit-chat is personalized, engaging, and of high quality. To this goal, we effectively integrate personalized information into a pre-trained language model UniLM [7], and propose a two-phase optimization schema for reinforced personalized text generation.

Finally, we conduct extensive experiments, including automatic quantitative experiments, case study, and user study, to verify the effectiveness and usefulness of our method.

2 USER STUDY ON CHIT-CHAT FOR NEWS RECOMMENDATION

We perform a user study to 1) understand *whether* external text like chit-chat can help increase a user's interest in recommended news and 2) provide a benchmark for evaluating whether a model can correctly capture a user's personalized interest towards a chit-chat.

To understand the impact of chit-chat on news recommendation, we collect 920 news articles published in Dec. 2019, and their corresponding 13,915 chit-chat posts in the subreddit "news" of the social platform Reddit. Then, we hire 50 participants from a data labeling company. For each participant, we randomly sample 200 or more pairs of chit-chat posts from the 13,915 posts, and provide them with both the chit-chat posts and the corresponding news headlines. The participants are required to answer three questions:

- Q1: Will you click the news after seeing the headline?
- Q2: Will you click the news after seeing the chit-chat post?
- Q3: Which chit-chat post can better attract you to click the news?

As shown in Table 1, chit-chats significantly increase the news click probability (+26%) and encourage users to read more (+27.6%).

The user study results can also be used for evaluating personalized chit-chat ranking models, since in Q3, we collect user preferences for different chit-chats. An issue is that these 50 participants cannot be directly linked to news recommendation datasets. To fill the gap, we directly collect which news articles the participants like. More details for the user study (e.g., screenshot for the labeling website) and data collection (e.g., data statistics and detailed collection process) are given in Appendix A.1.

3 PRELIMINARY

3.1 News Recommendation

News recommendation aims to predict a user's personalized interest in a news article based on the click history of the user. Specifically, a news recommendation dataset contains a list of users $\mathcal{U}$ and a set of news articles $\mathcal{N}$. A **user** $u \in \mathcal{U}$ is represented by the news articles that s/he clicked: $u = (n_1, n_2, ...n_{|u|})$, and each **news article** $n_i \in \mathcal{N}$ consists of a news headline and news body. We represent n_i as a sequence of sentences $n_i = [S_1, ..., S_{|n_i|}]$, where the first sentence is the headline h_i, and the other sentences are from the news body. Based on the dataset, a **news recommendation model** $f(u, n_i)$ can be learned to predict whether a user u will click a news article n_i. Most news recommendation models encode only the news headlines [34, 35], since users decide whether to click a news article based only on the headline, i.e., $f(u, n_i) = f(u, h_i)$.

3.2 Problem Formulation

The problem of generating personalized chit-chat for news recommendation can be defined as follows.

Model input. Given a news recommender system, the input of a personalized chit-chat generation model is a user-news pair (u, n_i), where $u \in \mathcal{U}$ and $n_i \in \mathcal{N}$ belong to the recommendation system.

Model output. The generation model outputs a chit-chat $\hat{c}$ tailored for the user u based on the news content n_i. The chit-chat $\hat{c}$ is represented by a sequence of word tokens: $\hat{c} = (\hat{y}_1, \hat{y}_2..., \hat{y}_{|\hat{c}|})$.

External textual corpus setting. Most existing works for personalized text generation assume that the type of text (e.g., reviews, or product descriptions) we wish to generate can be found in the recommendation dataset. However, this poses a high standard to the recommendation dataset, which typically does not contain text such as chit-chat. In this paper, we propose a novel setting in which an external textual corpus is used for generation.

Specifically, the chit-chat corpus we use contains a set of news articles $\mathcal{N}'$ and chit-chat posts C. The corpus can be disconnected with the recommendation dataset from both the user and item (news) perspective: we neither require that the users $\mathcal{U}$ can be found in the chat corpus, nor assume that the set of news in the chat corpus ($\mathcal{N}'$) overlaps with that in the recommendation dataset ($\mathcal{N}$). The only things that we require are:

- RQ1. The news articles in both datasets contain the same set of features (in our case, titles and bodies), and share a similar distribution (e.g., contain news articles on a similar set of topics);
- RQ2. There is an N-to-N mapping between news articles in $\mathcal{N}'$ and chit-chat posts in C. This means that we know roughly which chit-chat posts are relevant to which news articles. We denote this mapping as $m(n'_i, c_j)$, which is 1 (or 0) when n'_i is (or is not) relevant with c_j. Such relevance can be easily obtained from online forums such as Reddit, in which people chat under a post for a news article. In cases where such relevance is absent, we may distill them using heuristics such as content similarities.

4 PERSONALIZED INTEREST IN CHIT-CHAT

A key challenge in using an external chat corpus is to understand users' personalized interest in a chit-chat post. In this section, we introduce how we estimate this in the absence of ground truth.

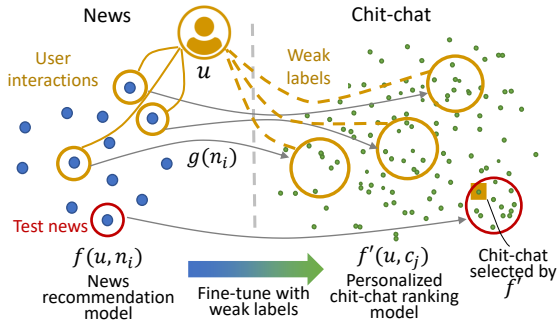

Figure 2: Weakly supervised paradigm for learning a personalized chit-chat ranking model.

4.1 Unsupervised Paradigm

A straightforward method is to directly use a news recommendation model to capture user interest. The basic idea is that any learned news recommendation model $f(u, n_i)$ that predicts how much user u likes news n_i could capture the personalized interest of users [1]. Thus, if we assume that the chit-chat and news headlines are in the same space, i.e., the news encoder learned in an existing news recommendation model could be used for accurately encoding chit-chat, then we could directly use f to measure users' personalized interest in a chit-chat post. In other words, we may measure the personalized interest of user u towards chit-chat post c_i by $f(u, c_i)$.

While this paradigm requires no supervised data and is easy to implement, it has two issues. First, $f(u, c_i)$ may not be accurate, since the chit-chat posts and news headlines are significantly different from multiple perspectives, ranging from wording and entities mentioned to language normativity and grammar structure. For example, in a chat corpus, fans usually refer to celebrities by using their nicknames (e.g., *Blondie* for *Taylor Swift*), which rarely appear in news articles. It is very difficult for the news encoder to correctly understand such entities (nicknames), which are important for capturing user interest. Second, this unsupervised paradigm cannot be extended to scenarios where items in the recommendation dataset contain other types of features, in addition to natural language ones. For example, the news may contain categorical features such as the publisher or editor of the news.

4.2 Weakly Supervised Paradigm

We solve the aforementioned problem by using a weakly supervised paradigm. Weak supervision refers to training with supervision signals that are incomplete (a small subset of labels), inexact (coarse-grained labels), or inaccurate (labels are not always ground-truth) [43]. Our scenario falls into the last category, in which we create a set of weak labels that may not always be accurate based on both the external chat corpus and the recommendation dataset.

As shown in Fig. 2, the basic idea is to use the external chat corpus to learn a model g that maps a news article into its corresponding chit-chat, and then use the mapped chit-chat posts to fine-tune the recommendation model f. In this way, the news ranking model f is changed to a chit-chat ranking model f', which replaces the previous news encoder with a chit-chat encoder, and solves the aforementioned issues of the unsupervised paradigm. For example,

g will change *Taylor Swift* in the news into *Blondie* in the chit-chat given an appropriate context, which helps f' to understand and encode nicknames such as *Blondie* correctly. Moreover, since f' is fine-tuned based on f, it is able to leverage the knowledge embedded in f, which is learned from the recommendation dataset and encodes users' personalized preferences. Given a large number of potentially interesting posts for an article, f' can be used to identify the one that best fits a user's interest (orange square in Fig. 2), based on its relations with the weak labels.

The key here is how to learn g, as well as how to generate and train with weak labels. Next, we first introduce a straightforward formulation and then show how the formulation can be refined based on the idea of crowd-sourcing [43].

4.2.1 Straightforward Formulation. To learn the model g that transfers news articles to chit-chat posts, a straightforward way is to directly use the N-to-N mapping m between the news articles and the chit-chat posts in the external chat corpus (RQ2 in Sec. 3.2). Specifically, g can be any natural language generation model, e.g., a pre-trained model UniLM [7]. We can learn g in a supervised manner with the cross-entropy loss $\mathcal{L}_{CE}$, so that it maximizes the likelihood of generating chit-chat c_j given a news article n'_i, if the chit-chat is relevant with the news (i.e., $m(n'_i, c_j) = 1$):

$$g = \arg\min_{g'} \mathcal{L}_{CE}(g') \tag{1}$$

$$\mathcal{L}_{CE}(g') = -\sum_{m(n'_i,c_j)=1} \log p(c_j = g'(n'_i)) \tag{2}$$

Weak labels. Given g, the labels for the personalized chit-chat ranking model f' could be obtained based on the formulation

$$f(u, n_+) > f(u, n_-) \Rightarrow f'(u, c_+) > f'(u, c_-) \tag{3}$$

$$\forall c_+ = g(n_+), c_- = g(n_-) \tag{4}$$

The underlying assumption is that, if a user prefers some news articles, then he will like the discussion and chit-chat about the corresponding news. The assumption is not always true, so it only allows us to derive weak labels. Accordingly, given a set of tuples (u, n_+, n_-) where n_+ (or n_-) is a news article the u clicked (or not clicked), a training sample of f' can be derived as $(u, g(n_+), g(n_-))$.

Note that g is trained with the news $\mathcal{N}'$ in the chat corpus (Eq. (2)) and applied in the news $\mathcal{N}$ in the recommendation dataset (Eq. (4)). Thus, generating accurate chit-chat in Eq. (3) requires that the two sets of news articles share the same set of features and a similar distribution (RQ1 in Sec. 3.2).

To make full use of the chat corpus to improve the ranking ability, we assume the user's reply behavior is a weak label of the user's click behavior. We regard the news the user replies to as the news clicked by the user and label the human-generated post according to if the user replies to the post. Therefore, besides transferring the knowledge of user interest on recommendation dataset using Eqs. (3) and (4), we can also learn the knowledge of user interest from the reply behavior on external chat corpus. The two data sources are in the same format and can be trained together.

Fine-tuning. Given weak labels $(u, c_+ = g(n_+), c_- = g(n_-))$, we can learn f' in the same way as any other recommendation model. Usually, a BPR loss is leveraged:

$$\mathcal{L}_{PER} = -\sum_{(u,n_+,n_-)} \log \frac{e^{f'(u,g(n_+))}}{e^{f'(u,g(n_+))} + e^{f'(u,g(n_-))}} \tag{5}$$

where f' reuses the architecture of f. f can be any news recommendation model. In this paper, we set f as NAML [35], which is a widely-used news recommendation model that well balances efficiency and accuracy.

4.2.2 Crowd-Sourcing-Based Learning. The weak labels obtained by using the straightforward formulation can be of low quality, since related chit-chat posts for a news article may contain a lot of noisy posts that are not informative, and can be quite diversified in terms of topic distribution. This causes two major problems. First, directly learning g with Eqs. (1) and (2) may easily result in sub-optimal results, e.g., generating generic chit-chat like "That's good news." [26]. Second, many training samples generated with g may fail to satisfy the assumption in Eq. (3): among a large number of posts for n_+, only a limited number of informative and relevant ones may be interesting to the user. Thus, in order to accurately capture user interest and learn a good f', it is essential that we improve g and the quality of generated weak labels.

To achieve this goal, we develop a crowd-sourcing-based formulation. Crowd-sourcing is a popular paradigm in learning with inaccurate supervision and has been widely used to collect labels in a cost-effective way [43]. Specifically, crowd-sourcing assigns potentially difficult labeling tasks to multiple low-quality workers and derives a good label by ensembling the noisy labels provided by the workers. Many methods have been developed to better ensemble the labels, such as majority voting and its variations [28], and a key of these algorithms is to decide the reliability of each worker.

Inspired by the idea of crowd-sourcing, we improve the quality of collected weak labels by learning multiple candidates $(g^{(1)}..., g^{(K)})$ for g, and treat each $g^{(k)}$ as an independent worker. Then, we compute relevance-based reliability for each g, and combine multiple weak labels with weighted fine-tuning. Specifically, our method contains the following three steps:

$g^{(1)}..., g^{(K)}$ as different workers. To improve the quality of generated chit-chat posts, we learn g so that it not only writes a post but also maximizes the relevance of the post with the news content. This enables us to avoid generating low-quality posts that are not informative or drifting away from the news content. Suppose there are K different relevance functions $\{Rel^{(k)}(n'_i, c_j)|k \in [1, K]\}$, we build $g^{(k)}$ for each of them:

$$g^{(k)} = \arg\min_{g'} \mathcal{L}^{(k)}_{CE}(g') + \lambda_e \mathcal{L}^{(k)}_{Rel}(g') \tag{6}$$

$$\mathcal{L}^{(k)}_{CE}(g') = -\sum_{\substack{m(n'_i,c_j)=1 \\ c_j = \arg\max Rel^{(k)}(n'_i,c_j)}} \log p(c_j = g'(n'_i)) \tag{7}$$

$$\mathcal{L}^{(k)}_{Rel}(g') = -\sum_{n'_i \in \mathcal{N}'} \sum_{c_j} Rel^{(k)}(n'_i, c_j)\ p(c_j = g'(n'_i)) \tag{8}$$

Compared with Eq. (2) which considers all posts, Eq. (7) filters noises by focusing only on the most relevant posts. Moreover, with Eq. (8), we can directly optimize relevance even if truly relevant posts are missing for some news. We use the pre-trained model UniLM [7] as the backbone for $g^{(k)}$, and utilize the reinforcement learning method UMPG [32] to efficiently and effectively optimize $g^{(k)}$.

Learning K candidates for g allows us to better handle imperfect relevance functions. Here, we adopt three common token-level relevance metrics and one additional deep neural relevance model to compute $Rel^{(i)}$. Specifically, we instantiate $Rel^{(1)}$ to $Rel^{(3)}$ with

three metrics: BLEU, ROUGE, and unigram F1 [19, 23]. To better capture the relevance in a data-driven way, we set $Rel^{(4)}$ to a relevance model $\hat{m}$ learned based on the N-to-N mapping m. In particular, we train the matching model fine-tuned on BERT [6]. We train a matching model using the following contrastive learning loss [9]:

$$\mathcal{L}_{CL}(\hat{m}) = -\sum_{m(n_i',c_+)=1,\ m(n_i',c_-)=0} log\frac{e^{\hat{m}(n_i',c_+)}}{e^{\hat{m}(n_i',c_+)}+e^{\hat{m}(n_i',c_-)}} \quad (9)$$

Since $Rel^{(4)} = \hat{m}$ is learned based on the N-to-N mapping m, it may be noisier compared with $Rel^{(1)}$ to $Rel^{(3)}$. However, it is able to capture some implicit relevance, e.g., the relevance between *Taylor Swift* and her nickname *Blondie* in the chit-chat.

Weak label derivation. Given multiple labels provided by different workers $g^{(1)}..., g^{(K)}$, we derive *soft* weak labels based on the reliability of each worker. Here, soft means that each weak label has a probability $p^{(k)}$ to be adopted:

$$f(u,n_+) > f(u,n_-) \Rightarrow f'(u,c_+^{(k)}) > f'(u,c_-^{(k)}) \text{ with } p^{(k)} \quad (10)$$

$$\forall c_+^{(k)} = g^{(k)}(n_+),\ \ c_-^{(k)} = g^{(k)}(n_-) \quad (11)$$

$$p^{(k)} = \bar{Rel}(n_+, c_+^{(k)})/\textstyle\sum_{k'} \bar{Rel}(n_+, c_+^{(k')}),\ k \in [1,K] \quad (12)$$

where $\bar{Rel}$ is an overall relevance model that measures the reliability of each worker. We find that simply setting $\bar{Rel}$ to $\hat{m}$ works sufficiently well empirically.

Weighted fine-tuning. Based on the soft weak labels, we can train f' with the following weighted loss function:

$$\mathcal{L}_{PER} = -\sum_{(u,n_+,n_-)}\sum_k p^{(k)} \log\frac{e^{f'(u,g^{(k)}(n_+))}}{e^{f'(u,g^{(k)}(n_+))}+e^{f'(u,g^{(k)}(n_-))}} \quad (13)$$

5 REINFORCED CHIT-CHAT GENERATION

In this section, we propose a reinforcement learning framework to generate Personalized chit-chat for news Recommendation using External Text corpora (PRET).

5.1 Reward Function

The reward function should evaluate both the attractiveness and the quality of a chit-chat post. Specifically, given a user u, a news article n_i, we design the reward function so that it scores a generated chit-chat post $\hat{c}$ from the following aspects:

$$\mathcal{R}(u,n_i,\hat{c}) = \mathcal{R}_{per}(u,\hat{c}) + \lambda_l\mathcal{R}_{like}(n_i,\hat{c}) + \lambda_r\mathcal{R}_{rel}(n_i,\hat{c}) + \lambda_c R_{chat}(\hat{c}) \quad (14)$$

Here, the first two rewards, $\mathcal{R}_{per}$ and $\mathcal{R}_{like}$, measure how attractive the chit-chat post is from personalized and non-personalized perspectives, respectively. The last two rewards, $\mathcal{R}_{rel}$ and $\mathcal{R}_{chat}$, evaluate the quality of the generated chit-chat posts based on whether they are relevant to the news article, and whether they fit the language model of chit-chat.

Personalized interest reward $\mathcal{R}_{per}$. We set $\mathcal{R}_{per}(u,\hat{c})$ to the score given by the personalized chit-chat ranking model f' learned in Sec. 4: $\mathcal{R}_{per}(u,\hat{c}) = f'(u,\hat{c})$. As shown in Fig. 2, among all posts that are suitable for a news article, f' helps select the one that best fits the click history of u.

Non-personalized attractiveness reward R_{like}. The attractiveness of a chit-chat post is also affected by non-personalized factors, such as whether it is informative and humorous. This is verified by the results of the user study in Sec. 2, which shows that jointly considering personalized user interest and non-personalized factors (e.g., number of votes) for a chit-chat post enables a more accurate prediction about user interest. Based on this observation, we add the "Updown" model in [9] as R_{like}, which predicts the number of up-votes of a Reddit post minus its number of down-votes and shows superior performance in predicting human preference.

Relevance reward $\mathcal{R}_{rel}$. To increase the probability that the user clicks news n_i, it is essential that the generated chit-chat is related to the news article. We measure this with the relevance model $\hat{m}$ trained with a contrastive learning loss in Eq. (9).

Chit-chat reward R_{chat}. We design R_{chat} to ensure that the generated $\hat{c}$ fits the language model of chit-chat. To ensure that the reward can be optimized efficiently during reinforcement learning, we follow Wang et al. [32] to compute R_{chat} based on whether a generated token matches the corresponding token in a pseudo-ground-truth chit-chat post. Here, the pseudo-ground-truth posts are generated with $g^{(k)}, k \in [1,K]$ (Eq. (6)). For one article n_i, we input K training samples, each with a different pseudo-ground-truth post.

5.2 Personalized UniLM as Generation Policy

The goal of the policy is to generate a personalized chit-chat tailored for user u based on news n_i. We adopt a Transformer-based pre-trained language model UniLM [7] as the backbone of our generation policy. The key question is how to inject personalized information into a pre-trained model effectively. Although user embedding v_u can be obtained from a news recommendation model f, directly injecting v_u into the pre-trained model can be problematic, since it usually does not align with the word embeddings [1]. For a pre-trained model, it is important that we properly integrate v_u so that 1) it does not interfere with existing model parameters, and at the same time 2) can be effectively leveraged to control the generation results. While there are works on integrating personalized information into a text generation model [1, 16, 33], how to integrate personalized information into pre-trained models effectively is under-explored.

To solve this issue, we propose a personalized UniLM model (Fig. 3). This model effectively integrates personalized information by combining the advantages of retrieval models. The basic idea is to first retrieve the most important tokens and sentence from the news content based on user embedding v_u and recommendation model f. This is a much easier task compared with directly generating personalized chit-chat, especially considering that the recommendation model f is already learned to correctly rank news. Then, the retrieved embedding z that aligns with word embeddings and the corresponding sentence are injected into the generation model to provide personalized information without interfering with the model parameters of UniLM. The $(t+1)$-th output token is sampled from the distribution of the generation embedding o_t:

$$p_{\text{gen}}(\hat{y}_{t+1}|u,n_i,\hat{y}_{\le t}) = \text{softmax}(W_1^T o_t) \quad (15)$$

where W_1 is the parameter to be learned. Next, we introduce in detail how we obtain the retrieved embedding z and the generation embedding o_t.

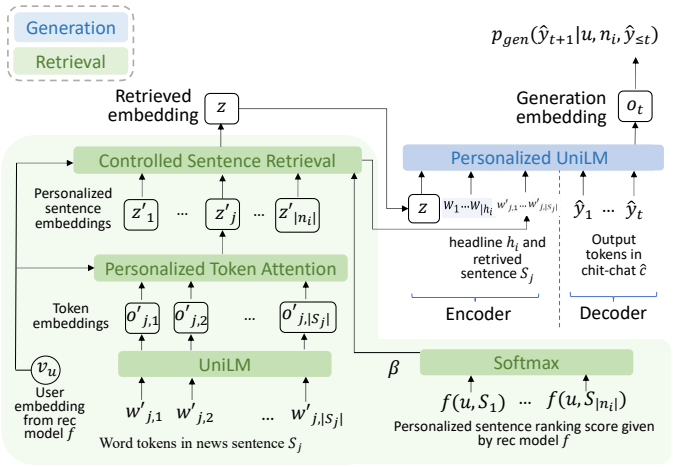

Figure 3: Personalized UniLM as the policy network.

Retrieval. As shown in Fig. 3, we retrieve z with three steps:
Step 1. UniLM as the contextualized token encoder. Given each sentence $S_j = [w'_{j,1}, w'_{j,2}..., w'_{j,|S_j|}]$ in news n_i, we compute the contextualized token embedding $o'_{j,l}$ for each word token $w'_{j,l}$ in the sentence by UniLM. This enables us to obtain embeddings that align with the word embeddings in the UniLM model.
Step 2. Personalized token attention. We then compute u's personalized attention γ_l on the tokens, and aggregate the embeddings of the important tokens to derive a personalized sentence embedding z'_j:

$$\gamma_l = \text{softmax}(w_2^T [v_u; o'_{j,l}]) \tag{16}$$

$$z'_j = \textstyle\sum_l \gamma_l \, o'_{j,l} \tag{17}$$

where v_u is the representation of user u learned in the news recommendation model f.
Step 3. Controlled sentence retrieval. Next, we retrieve important sentences based on personalized sentence attention. Following Lian et al. [18], we compute a hard attention $\beta_j \in \{0, 1\}$ with Gumbel-Softmax [10], which eliminates noise and simplifies the information we give the generator to control text generation better:

$$\beta_j = \text{Gumbel\_softmax}(w_3^T [v_u; z'_j]) \tag{18}$$

$$z = \textstyle\sum_j \beta_j z'_j \tag{19}$$

where z is the retrieved embedding that aligns with the word embeddings in UniLM. To facilitate sentence retrieval, we pre-train the model by leveraging the recommendation model f. Specifically, for each sentence S_j, we estimate how much user u likes the sentence with $f(u, S_j)$, and treat it as a prior importance score for the sentence. Then, we normalize the prior importance to obtain a probability distribution p_o, and compare p_o with the probability p_r for retrieving a sentence in personalized UniLM with KL divergence:

$$\mathcal{L}_{KL} = D_{KL}(p_r|p_o) = \textstyle\sum_j p_r(S_j) \log \frac{p_r(S_j)}{p_o(S_j)} \tag{20}$$

$$p_o(S_j) = \text{softmax}(w_3^T [v_u; z'_j])) \tag{21}$$

$$p_r(S_j) = \text{softmax}(f(u, S_j)) \tag{22}$$

Besides this prior importance, the posterior reward obtained after generating the post can also be used to train the sentence retrieval module. We adapt policy gradient with baseline.

$$\mathcal{L}_{rl} = -(R(u, n, \widetilde{c}) - R(u, n, \hat{c})) \log p_r(\widetilde{s}) \tag{23}$$

where sentence $\widetilde{s} \in S$ is sampled according to p_r, the baseline sentence is $\hat{s} = \arg\max_{s \in \mathcal{S}} p_r(s)$. $\widetilde{c}$ and $\hat{c}$ are the posts generated with $\widetilde{s}$ and $\hat{s}$ separately.

Generation. After retrieving, we input z to UniLM, together with the word tokens $[w_1, w_2..., w_{|h_i|}]$ in the news headline h_i, selected sentence $[w'_{j,1}, w'_{j,2}, ..., w'_{j,|S_j|}]$, as well as the previously decoded tokens $\hat{y}_1..., \hat{y}_t$. The embedding of $\hat{y}_t$ in the last layer of UniLM is then considered as the generation embedding o_t.

6 OPTIMIZATION

We summarize the optimization of the above modules into a two-phase optimization schema.

Phase 1. Non-personalized chit-chat generation. We first fine-tune UniLM using the external chat corpus. This enables us to learn a non-personalized model that can generate relevant and fluent chit-chat posts for a news article. Then, this model is used to train different workers described in Sec. 4 using Eqs. (7) and (8).

Phase 2. Personalized chit-chat generation with reinforcement learning. In this phase, we inject the personalized information. First, we integrate different workers to train the model for predicting the personalized interest reward $\mathcal{R}_{per}$ using Eq. (13). Then, we use the non-personalized chit-chat generation model as the base model to train the personalized UniLM model with the reinforcement learning method UMPG [32] to maximize the reward $\mathcal{R}$ (Eq. (14)). We optimize the generation module following UMPG and optimize the retrieve module using Eqs. (20) and (23). We use UMPG here because it can be integrated seamlessly with pre-trained language models, and largely increase the training efficiency.

7 EXPERIMENT

In this section, we first conduct experiments to evaluate the performance of the proposed model estimating the users' personalized interest in a chit-chat post. Then we compare our proposed PRET with other chit-chat generation methods.

7.1 Datasets

We use two datasets in the experiments. **MIND** is a large dataset for news recommendation, and it contains the impression logs of the users. Each impression includes a user u, a set of news n, and the associated labels representing if the news is clicked by the user. **Reddit** is a large social platform which has been widely used as a natural data corpus for open-domain conversation. For each news, it is associated with a set of user-generated chit-chat posts. Please see more details to construct this dataset in Appendix A.2.

The statistical results of the two datasets are detailed in Table 7.

7.2 Evaluation Metrics

To evaluate the personalized chit-chat ranking model, we follow the assumption in Eqs. (3) and (4) to construct the chit-chat posts to rank, and use three commonly-used ranking metrics **AUC**, **MRR**, and **NDCG@5** [11, 30]. We also use **AUC** as the metric to test

the performance of our proposed personalized interest model with several chit-chat scoring models and study the factors that affect user interest in our proposed user study dataset.

To evaluate the performance of the generation models, we first evaluate the degree of personalization. For each news, we generate the chit-chat posts for 5 users and compute the score of the generated posts by the personalized interest model **Per**. We also compute the token-level **Distinct** score as another personalization metric, i.e., evaluating if the model can generate diverse posts for different users or only learn knowledge of popularity to generate the same post for different users. Then we evaluate the generation quality. As there is no ground-truth chit-chat post provided in MIND, we adopt model-based evaluation trained on a large chit-chat corpus [9], including **Updown**, **Depth**, and **Width**, which has a higher human preference correlation than ppl. and BoW baseline.

To evaluate the relevance, we use the **Relevance** score predicted by (Eq. 9). As we directly optimize the user interest score in this framework, we also conduct a user study to validate the usefulness of user interest from four aspects.

7.3 Compared Methods

7.3.1 Methods for Quantifying User Interest.

We evaluate the ability to quantify the user interest for both non-personalized and personalized interest models.

Non-personalized interest model. We adopt two models **Updown** and **Width** proposed by [9] as the non-personalized baselines. The two models are chit-chat ranking models trained on Reddit using the human-feedback information as the labels.

Personalized interest model. We evaluate our proposed three training strategies of personalized interest model on different backbone models.

First, we recall the training strategies we use. **Rec** is using the MIND dataset to train the recommendation models, which is an unsupervised method for chit-chat ranking. **Single Worker** is the weakly supervised method with only one worker and it is named Single Worker. **Crowd-Sourcing** is the weakly supervised method with the ensemble mechanism to use the labels provided by multi workers and it is named as Crowd-Sourcing.

Then, for the backbone models, We choose NAML and NRMS [34, 35], which are two methods commonly used in news recommendation tasks. To enhance the language understanding, we also use a pre-trained language model as the news encoder. We combine the two models with the three training methods, respectively.

7.3.2 Methods for Personalized Chit-chat Generation. To compare the generation ability, we also compare our proposed model with non-personalized and personalized baselines.

Non-personalized baselines: For non-personalized baselines, we adopt two common transformer-based generation models. **Transformer** is a widely-used baseline for seq2seq generation task [29]. **UniLM** is a pre-trained generation model fine-tuned on our Reddit corpus [7].

Personalized baselines: We compare our proposed personalized model with two baselines of different methods to inject the personalized information. **PENS-UniLM** is using PENS to generate a personalized news headline [1] as the personal information and then feeding it into UNILM. **RL-Vec** is directly adding a user vector learned from a news recommendation model to the input of UniLM instead of using our proposed content retrieve methods. It is optimized using the reinforcement learning method UMPG to maximize the reward $\mathcal{R}$ (Eq. (14)). **PRET** is our proposed framework. It consists of retrieval and generation modules to generate chit-chat posts.

Models	chit-chat			MIND		
	AUC	MRR	NDCG@5	AUC	MRR	NDCG@5
NAML-Rec	61.71	29.24	32.06	**69.32**	**34.32**	37.86
NAML-Single Worker	64.64	31.05	34.03	69.27	34.19	**37.88**
NAML-Crowd-Sourcing	**66.29**	**31.87**	**34.73**	69.16	34.19	37.76
NRMS-Rec	63.34	29.62	32.28	67.88	32.98	36.41
NRMS-Single Worker	65.15	31.08	33.96	68.29	33.32	36.98
NRMS-Crowd-Sourcing	65.44	31.08	33.80	68.39	33.43	37.04

Table 2: The evaluation results of personalized user interest on MIND.

Ensemble				AUC	
Updown	Width	Rec	Per	Clicked	Unclicked
✓	-	-	-	0.568	0.550
-	✓	-	-	0.558	0.549
-	-	✓	-	0.534	0.544
-	-	-	✓	0.547	0.558
✓	✓	-	-	**0.571**	0.566
✓	✓	-	✓	0.570	**0.577**

Table 3: The evaluation results of personalized user interest on the user study dataset.

7.4 Evaluation Results

7.4.1 Quantifying User Interest.

In Table 2, we compare our proposed methods with the unsupervised baseline. The column of "chit-chat" lists the results of chit-chat post ranking. The unsupervised model is not able to distinguish the user interest in the chit-chat post thoroughly, as it lacks the knowledge of the mapping of news and chit-chat posts, while the weakly supervised methods Single Worker and Crowd-Sourcing can alleviate this issue and give rise to the AUC score on NAML and NRMS. The Crowding-Sourcing method brings the gain of 5.26 and 2.1 of AUC on NAML and NRMS, respectively. Comparing Single Worker and Crowd-Sourcing, we can see that the Crowd-Sourcing methods can effectively reduce the noise and perform better. In the following section, we use **Per** to denote NAML-Crowd-Sourcing. In the column of "MIND", we report the evaluating results of the original news recommendation task on MIND.

We further test the model on our proposed user study dataset. It is a harder ranking task as the candidates are related to the same news and the user interest contains multi-aspects, e.g., the chatting style and the personalized news interest. The results in Table 3 show that the non-personalized interest model and personalized interest model have comparable performances. The engagement of chit-chat is related to both non-personalized factors that could be quantified with the count of likes and replies, as well as personalized factors that vary for different users. The ensemble results bring more gain, which demonstrates the importance of considering multiple factors. We also report the results for clicked news and unclicked news. We can see the non-personalized interest is more important to the news the user likes and the personalized interest is more important for the news that the user does not like originally.

Models	Personalized interest		Non-personalized interest			Relevance
	Per	Distinct	Updown	Depth	Width	
Transformer	0.686	0.2	0.290	0.367	0.417	0.746
UniLM	0.719	0.20	0.381	0.394	0.419	0.869
PENS-UniLM	0.725	0.33	0.369	0.414	0.442	0.812
RL-Vec	0.740	0.23	0.392	0.492	0.530	0.890
PRET	**0.769**	**0.42**	**0.512**	**0.520**	**0.593**	**0.913**

Table 4: The evaluation results of personalized chit-chat generation on MIND.

Metrics	PRET vs. UniLM			PRET vs. Transformer		
	Win	Tie	Loss	Win	Tie	Loss
Click	31.1%	41.6%	27.2%	43.9%	45.8%	10.2%
Relevance	34.9%	28.2%	26.7%	46.3%	43.6%	9.94%
Fluency	35.7%	32.6%	31.5%	54.2%	36.4%	9.35%
Informative	31.1%	45.0%	23.8%	39.8%	51.3%	8.78%

Table 5: Pair-wise human judgment.

7.4.2 Personalized Chit-chat Generation.

From the results of Table 4, the Transformer and UniLM get 0.2 for the Distinct score, which is the lower bound of this metric as we distribute 5 users for each news. The results of Transformer and UniLM indicate that without pre-trained knowledge or other effective training methods, the Transformer tends to generate generic and dull chit-chat. For the results of the personalized model, the RL-Vec model can get a higher personalized score than UniLM and Transformer. But the diversity score is still low, which means it mainly learns the popularity part of the recommendation model but ignores the information of the user vector. The PENS-UniLM achieves a better diversity and personalized score than the unsupervised baseline. It shows the generated headlines can benefit personalization. But the personalized score is lower than RL-Vec and PRET. They are two possible reasons. First, the two-stage method loses the user information as UniLM does not see the user information directly. Second, the generated headline tends to cover the content of the whole body, which limits the diversity. Our proposed PRET outperforms the baselines from personalized interest and non-personalized interest. From the result of Table 5, PRET not only increases the click rate but also appeals to the user from the dimension of relevance, fluency, and informative score.

The analysis of the pros and cons of PERT through a case study is available in Appendix C.2 and we conduct an ablation study on the retrieve module and user interest estimation in Appendix B.

8 RELATED WORK

8.1 Personalized Text Generation for Recommendation

Many methods have been proposed to generate personalized text to facilitate recommendation, such as textual explanation generation for recommendation [3, 4, 16, 39], E-commerce review summarization [36], product description generation [33], and news headline generation [1]. Pioneering works in this line often output text by filling predefined templates [8, 14, 40] or retrieving from the existing text (e.g., user reviews) [31]. Recently, researchers have focused more on generating personalized text word by word using deep-learning-based methods, in order to avoid rigid templates, better fit users' personalized interests, and eliminate copyright issues [4].

A key question in personalized text generation is how to integrate the personalized information into a natural language generation model. A popular way is to use a multi-task framework that generates personalized text and makes recommendation simultaneously [4, 16]. Chen et al. [3] further improve this framework to generate explainable responses and recommend items in an interactive manner. Li et al. [15], Xu et al. [36] adjust the Transformer structure. Wang et al. [33] adopt a reinforcement learning method to generate personalized product descriptions. Ao et al. [1] incorporate a user vector to generate headlines. While existing methods achieve certain success, they typically assume that the type of text to be generated (e.g., reviews or headlines) can be found in the recommendation dataset. This prevents them from being applied in scenarios where a desirable textual corpus does not exist in the recommender system. We bridge the research gap by showing how external textual corpora could be leveraged for generating personalized text with a weakly supervised paradigm. Moreover, how we can integrate personalized text into a pre-trained language model is under-explored. To solve this issue, we design a reinforcement learning framework for generating personalized chit-chat based on the pre-trained language model UniLM.

8.2 Conversational Recommendation

Conversational recommendation aims to obtain the preference of users during conversation and/or make persuasive recommendation. Some works model the dialog strategy of which attribute to ask and what item to recommendation [5, 27]. They usually simulate the conversation and adopt templates for creating responses. In order to generate more context-relevant responses, some works train a response generator based on the crowd-sourced conversational recommendation data [17]. In order to generate more proper responses, Chen et al. [2], Zhou et al. [42] leverage knowledge graphs or reviews as external knowledge. Our method also deals with the response generation task, but we generate a response to arouse users' interest so that they will have a more engaging news discussion and news reading experience rather than explicitly persuading the user to click or buy the item. Note that our method can benefit the attractiveness of generated responses for conversational recommendation systems.

8.3 Personalized Dialog Generation

Methods of personalized dialog generation can be divided into two categories based on whether they enable personalization for the chatbot or the end-users. The first category makes the chatbot more human-like by assigning them a consistent personality of different forms, such as key-value attributes [22], detailed description sentences [38] and chatting habits [21, 25]. The second category focuses on modeling the preference of users [12, 24, 37], i.e., generating various responses that are tailored to the profile of the end-users [12]. Based on the dataset, some methods have been proposed, and most of them adopt different variants of Memory Network [24, 37] for modeling user preference. Our task falls into the second category. While existing methods rely on explicit user attributes, e.g., age and gender, our method can additionally leverage the implicit user preference learned from the user-item interactions. This allows our method to generate conversations that are better tailored to user

tastes and utilize advanced recommendation models for improving user engagement.

9 CONCLUSION

In this paper, we conduct a user study and collect a dataset to verify the effectiveness of chit-chat in increasing the probability that a user reads a recommended news article. We also propose PRET to generate personalized chit-chat for news recommendation and design a weakly supervised method for estimating users' personalized interest in a chit-chat post. Extensive experiments show our approach can generate personalized chit-chat posts and attract the user to click the news.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (NSFC Grant No. 62122089 and No. 61876196), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. We also wish to acknowledge the support provided and contribution made by Ying Qiao from Microsoft News, Professor Xiang Ao and Zhao Yang from Chinese Academy of Sciences, as well as Public Policy and Decision-making Research Lab of RUC. Rui Yan is supported by Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. 2021. PENS: A Dataset and Generic Framework for Personalized News Headline Generation. In *ACL*. 82–92.
[2] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. In *EMNLP-IJCNLP*. 1803–1813.
[3] Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. 2020. Towards Explainable Conversational Recommendation. In *IJCAI*. 2994–3000.
[4] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *IJCAI*. 2137–2143.
[5] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *SIGIR*. 1431–1441.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.
[7] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeuIPS*. 13063–13075.
[8] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable recommendation through attentive multi-view learning. In *AAAI*, Vol. 33. 3622–3629.
[9] Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and William B Dolan. 2020. Dialogue Response Ranking Training with Large-Scale Human Feedback Data. In *EMNLP*. 386–395.
[10] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).
[11] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (oct 2002), 422–446.
[12] Chaitanya K. Joshi, Fei Mi, and Boi Faltings. 2017. Personalization in Goal-Oriented Dialog. *ArXiv* abs/1706.07503 (2017).
[13] Zhiyu Kong, Xiaoru Zhang, and Ruilin Wang. 2021. Review of the Research on the Relationship Between Algorithmic News Recommendation and Information Cocoons. In *ICLAHD*. 341–345.
[14] Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate neural template explanations for recommendation. In *CIKM*. 755–764.
[15] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized Transformer for Explainable Recommendation. In *ACL*. 4947–4957.
[16] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR*. 345–354.
[17] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *NeuIPS*, Vol. 31.
[18] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *IJCAI*. 5081.
[19] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).
[20] Danyang Liu, Jianxun Lian, Zheng Liu, Xiting Wang, Guangzhong Sun, and Xing Xie. 2021. Reinforced Anchor Knowledge Graph Generation for News Recommendation Reasoning. In *SIGKDD*. 1055–1065.
[21] Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. One Chatbot Per Person: Creating Personalized Chatbots based on Implicit User Profiles. In *SIGIR*. 555–564.
[22] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. In *EMNLP*. 2775–2779.
[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*. 311–318.
[24] Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2021. A cooperative memory network for personalized task-oriented dialogue systems with incomplete user profiles. In *Proceedings of the Web Conference 2021*. 1552–1561.
[25] Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, Yutao Zhu, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. 2021. Pchatbot: A Large-Scale Dataset for Personalized Chatbot. In *SIGIR*. 2470–2477.
[26] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, Vol. 30.
[27] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *SIGIR*. 235–244.
[28] Tian Tian and Jun Zhu. 2015. Max-Margin Majority Voting for Learning from Crowds.. In *NeuIPS*. 1621–1629.
[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeuIPS*, Vol. 30.
[30] Ellen M. Voorhees. 1999. The TREC-8 Question Answering Track Report. In *TREC*.
[31] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. *ICDM* (2018), 587–596.
[32] Xiting Wang, Xinwei Gu, Jie Cao, Zihua Zhao, Yulan Yan, Bhuvan Middha, and Xing Xie. 2021. Reinforcing Pretrained Models for Generating Attractive Text Advertisements. In *SIGKDD*. 3697–3707.
[33] Yongzhen Wang, Jian Wang, Heng Huang, Hongsong Li, and Xiaozhong Liu. 2020. Evolutionary Product Description Generation: A Dynamic Fine-Tuning Approach Leveraging User Click Behavior. In *SIGIR*. 119–128.
[34] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. *IJCAI* (2019).
[35] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*. 6389–6394.
[36] Hongyan Xu, Hongtao Liu, Pengfei Jiao, and Wenjun Wang. 2021. Transformer Reasoning Network for Personalized Review Summarization. In *SIGIR*. 1452–1461.
[37] Bowen Zhang, Xiaofei Xu, Xutao Li, Yunming Ye, Xiaojun Chen, and Zhongjie Wang. 2020. A memory network based end-to-end personalized task-oriented dialogue generation. *Knowl. Based Syst.* 207 (2020), 106398.
[38] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too?. In *ACL*. 2204–2213.
[39] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.* 14, 1 (mar 2020), 1–101.
[40] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*. 83–92.
[41] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In *SIGIR*. 239–248.
[42] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *SIGKDD*. 1006–1014.
[43] Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National science review* 5, 1 (2018), 44–53.

SUPPLEMENTARY MATERIAL

A DATASET COLLECTION

A.1 User Study Details

We collect 920 news articles published in Dec. 2019, and their corresponding 13,915 chit-chat posts in the subreddit "news" of the social platform Reddit. The statistical results is listed in Table 8.

We pre-process around 10,000 pairs of chit-chat posts and ask 50 annotators to label the chit-chat posts. These annotators are hired via a data labeling company. All of them are English native speakers and have experience in using at least one news reading platform. In the first stage, the annotators are asked to choose at least 50 news headlines from 1,000 given headlines. The 1,000 headlines are sampled from the news from Sept. 2019 to Nov. 2019 on Reddit. In the second stage, for each participant, we randomly sample 200 or more pairs of chit-chat posts from the 13,915 posts, and provide them with both the chit-chat posts as well as the corresponding news headlines. The participants are required to answer the following three questions:

- Q1: Will you click the news after seeing the headline?
- Q2: Will you click the news after seeing the chit-chat post?
- Q3: Given the pair of chit-chat posts, Which one can better attract you to click the news?

We give an example of the page we are using for the user study in Fig. 4, where each annotator is presented with news headlines, chit-chat posts and several questions.

A.2 Reddit Dataset

We collect the headline and posts From Jan. 2018 to Dec. 2019 from the directories "comments" and "submissions" in https://files.pushshift.io/reddit/. We keep the posts in the subreddit of "news" and "worldnews" and filter posts with dirty words or posts written by bots using the code provided by https://github.com/nouhadziri/THRED. There are key-value data to describe the attributes of each post, e.g., "body", "author", "score", and "parent_id". The "parent_id" provides the replying relations in headline-post pairs and post-post pairs. Thus it can expand to a tree structure according to the replying relations with the headline as the root. Each path from the root to one node can form the dialogue history. The children of each node can be regarded as multiple references for predicting response given the dialogue history. The non-personalized chit-chat generation model is trained on the dialogue history and response. To collect the personalized reply behaviour of users, we take advantage of the field of "author" of each post, i.e., collect the posts of the same "author" and extract the corresponding news to represent his personalized interest. At last, to collect the news body, as the key-value data only provides the headline and its URL, we use a crawler[3] to get the news content through the URL of the headline.

B ABLATION

B.1 Ablation Study on Retrieve Module

To study how the sentence retrieved can impact the performance, we conduct ablation on the module of Controlled Sentence Retrieval. The results from Table 6 show the effectiveness of the sentence retrieval method. After selecting randomly or selecting the least preferred content, the score of the personalized interest decrease. After removing the supervision of the prior importance, the diversity decreases. This phenomenon indicates the agent chooses to exploit the popular content rather than explore the personalized part. This may suffer from the personalized preference sparsity, i.e., the sampled user might not have a preference for the news content. The agent can only remember the choice of popular content. As the model benefits from the prior importance score, we also set the experiment to directly feed the sentence with the maximum prior importance into the generation part. We can see PRET still outperforms this strong variant. It demonstrates the RL framework can learn more knowledge beyond the prior.

[3]https://pypi.org/project/newspaper3k/

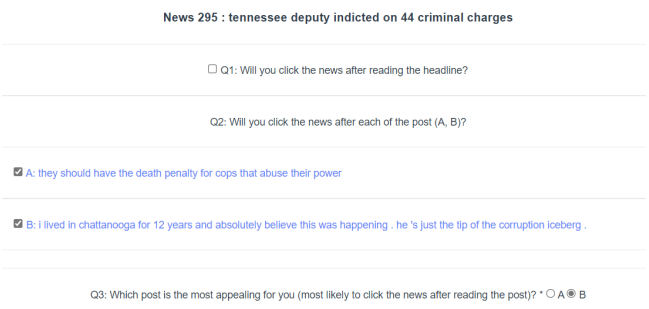

Figure 4: An example of the user study.

Models	Personalized interest		Non-personalized interest			Relevance
	Per	Distinct	Updown	Depth	Width	
PRET	**0.769**	0.42	**0.512**	**0.520**	**0.593**	**0.913**
PRET($content_{prior}$)	0.765	0.48	0.505	0.518	0.588	0.907
PRET($content_{min}$)	0.710	0.40	0.491	0.488	0.568	0.910
PRET($content_{random}$)	0.731	**0.60**	0.482	0.497	0.562	0.906
PRET($-Prior\ Importance$)	0.758	0.23	0.444	0.429	0.476	0.896

Table 6: The ablation study of PRET on MIND dataset.

	#news	#post	#impression	#user	Avg. body len.	Avg. title len.
MIND	161,013	-	15,777,377	1,000,000	585.05	11.52
Reddit	65,399	665,298	-	-	715.09	13.33

Table 7: The statistical results of MIND and Reddit.

#news	#post	#user	Avg. post len.	Avg. title len.
920	13,915	50	16.6	14.23

Table 8: The statistical results of the user study dataset.

B.2 Ablation Study on User Study Dataset

We do another ablation study to see how the weak labels of reply behavior of the Reddit dataset affect the performance of estimating user interest. The result is shown in Table 11. Per(*rand*) means the negative sample is sampled randomly from other news on Reddit. Per(*same*) means the negative sample is sampled randomly from the same news on Reddit. We can see that Per(*same*) outperforms Per(*rand*) when evaluated individually. But after ensembled with Updown and Width, Per(*same*) doesn't bring more gain to the overall performance. One possible reason is that Per(*same*) learns more about non-personalized interests than Per(*rand*). As the candidates are from the same news, the model needs to rank the candidates by non-personalized interests for the optimization. As a result, the

Clicked news	Southwest pilots sue Boeing for $100 million over lost wages from 737*MAX*	
Headline	Boeing CEO Dennis Muilenburg to step down immediately	
Post1	i 'll still never fly on a 737*max*	☑
Post2	weird how a ceo always "steps down" and is never "fired"	☐
Clicked news	Why Giving Up Meat Won't Have Much of an Effect on *Climate* Change	
	Gore kicking off 24 hours of climate talks around the world	
Headline	Soccer Legend Megan Rapinoe Named Sports Illustrated's Sportsperson Of The Year	
Post1	and "*climatestrike*" was named word of the year. i'm sensing a trend here...	☑
Post2	how is she a legend? maybe she should stick to soccer and keep politics out of it	☐

Table 9: Cases of the user study dataset.

Headline	NBA Finals 2022 – Complete News, Schedules, Stats for Golden State Warriors vs. Boston Celtics
Selected sentence	the warriors, led by western conference finals mvp stephen curry, are in the finals for the sixth time
Generation	so what about the warriors? i mean, it' s nice to see them win.
Selected sentence	boston hasn' t won the title since 2008, and no one on the celtics roster has ever played
Generation	til that boston has never won a championship. source: am celtics fan.

Table 10: Cases of generated posts.

Ensemble				AUC	
Updown	Width	Per(*rand*)	Per(*same*)	Clicked	Unclicked
-	-	✓	-	0.547	0.558
-	-	-	✓	0.556	0.559
✓	✓	-	-	**0.571**	0.566
✓	✓	✓	-	0.570	**0.577**
✓	✓	-	✓	0.568	0.567

Table 11: The ablation study of estimating user interest on the user study dataset.

model learn similar knowledge with Updown and Width and can not bring gain through ensembling.

C CASE STUDY

C.1 User Study Dataset

We give cases of the results of the annotation in Table 9. The checkbox shows the choices of the annotators of Q3 in Appendix A.1, where the checked box represents the post the annotator prefers. The two annotators may be interested in the topics of '737 MAX" and 'Climate' separately and choose the corresponding chit-chat post.

C.2 Case Study of Generation Results

In Table 10, we give examples of the retrieved sentences and the generation results based on the sentences. The example shows a typical application of the sentence retrieval module in the field of competitive sports news. As different users support different teams, the model can select the sentences related to the team the user like in the news to generate personalized chit-chat posts. As we inject an external user vector for the sentence selection, the personalization of the model mainly relies on the sentence retrieval module and some semantics of the user vector cannot be fully utilized, such as mentioning the news that the user has seen.

For the generation module, we can find from the case that the model mainly has two generation modes. One is to summarize the sentence, while another is to generate comments that look like personal views. For the first mode, the summarization is closer to the selected sentences, in line with the facts and prevents generating nonsense posts. For the second mode, from the given cases in Table 9, we can know providing personal views may bring more interest. But the generation module is hard to generate a post as interesting as the human-generated post. On the one hand, the model may generate counterfactual sentences. On the other hand, since generating personal opinions may require the support of other external knowledge other than news content, the novelty of the generated personal view is also lower than that shown in the user study. There can be a compromise between the two modes and more external knowledge may be taken into account for the generation of the post.