PP-Rec: News Recommendation with Personalized User Interest and Time-aware News Popularity

Tao Qi[1], Fangzhao Wu[2], Chuhan Wu[1] and Yongfeng Huang[1]

[1]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, Beijing 100080, China
{taoqi.qt, wufangzhao, wuchuhan15}@gmail.com
yfhuang@mail.tsinghua.edu.cn

Abstract

Personalized news recommendation methods are widely used in online news services. These methods usually recommend news based on the matching between news content and user interest inferred from historical behaviors. However, these methods usually have difficulties in making accurate recommendations to cold-start users, and tend to recommend similar news with those users have read. In general, popular news usually contain important information and can attract users with different interests. Besides, they are usually diverse in content and topic. Thus, in this paper we propose to incorporate news popularity information to alleviate the cold-start and diversity problems for personalized news recommendation. In our method, the ranking score for recommending a candidate news to a target user is the combination of a personalized matching score and a news popularity score. The former is used to capture the personalized user interest in news. The latter is used to measure time-aware popularity of candidate news, which is predicted based on news content, recency, and real-time CTR using a unified framework. Besides, we propose a popularity-aware user encoder to eliminate the popularity bias in user behaviors for accurate interest modeling. Experiments on two real-world datasets show our method can effectively improve the accuracy and diversity for news recommendation.

1 Introduction

Personalized news recommendation is a useful technique to help users alleviate information overload when visiting online news platforms (Wu et al., 2020d,b, 2021; Ge et al., 2020). Existing personalized news recommendation methods usually recommend news to a target user based on the matching between the content of candidate news and user interest inferred from previous behaviors (Zhu et al., 2019; Wu et al., 2019f). For example, Wu et al.

Figure 1: Several example popular news.

(2019e) proposed to model news content from news title based on multi-head self-attention. In addition, they modeled user interest from the previously clicked news articles with multi-head self-attention to capture the relatedness between different behaviors. An et al. (2019) proposed to use CNN network to learn news embeddings from news titles and categories, and model both long-term and short-term user interests from news click behaviors. However, these personalized news recommendation methods usually have difficulties in making accurate recommendations to cold-start users, since the behaviors of these users are very sparse and it is difficult to model their interest (Trevisiol et al., 2014). Besides, these methods tend to recommend similar news with those users have read (Nguyen et al., 2014), which may hurt user experience and is not beneficial for them to receive new information.

The motivation for this work is that popular news usually convey important information such as catastrophes, epidemics, presidential election and so on, as shown in Fig. 1. These popular news can attract many users to read and discuss even if they have different personal interest (Yang, 2016). In addition, popular news are diverse in content and can cover many different topics (Houidi et al., 2019). Thus, incorporating popular news has the potential to alleviate the cold-start and diversity problems in personalized news recommendation.

In this paper, we propose a new method named *PP-Rec* for news recommendation[1], which can consider not only personalized user interest in news but also the popularity of candidate news. In our method, the ranking score of recommending a candidate news to a target user is the combination of a personalized matching score and a news popularity score. The personalized matching score is used to measure personal user interest in the content of candidate news. The news popularity score is used to measure the time-aware popularity of candidate news. Since news popularity is influenced by many different factors such as content and freshness, we propose a unified model to predict time-aware news popularity based on news content, recency, and near real-time click-through rate (CTR). These two scores are combined via a personalized aggregator for news ranking, which can capture the personalized preferences of different users in popular news. Moreover, we propose a knowledge-aware news encoder to generate news content embeddings from both news texts and entities. Besides, since news popularity can effect users' click behaviors (Zheng et al., 2010) and lead to bias in behavior based user interest modeling, we propose a popularity-aware user encoder which can consider the popularity bias in user behaviors and learn more accurate user interest representation. Extensive experiments on two real-world datasets show *PP-Rec* can effectively improve the performance of news recommendation in terms of both accuracy and diversity.

2 Related Work

2.1 Personalized News Recommendation

Personalized news recommendation are widely used in online news platforms (Liu et al., 2010; Bansal et al., 2015; Wu et al., 2020d,c, 2019d). Existing personalized news recommendation methods usually rank candidate news for a target user based on the matching between news content and user interest (Wang et al., 2018; Qi et al., 2020; Wu et al., 2020a, 2019c). For example, Okura et al. (2017) learned news embeddings from news bodies via an auto-encoder and modeled user interests from the clicked news via a GRU network. The matching between news and user is formulated as the dot product of their embeddings. Wu et al. (2019e) used multi-head self-attention networks to generate

news content embeddings from news titles and generate user interest embeddings from clicked news. They also used the dot product of user and news embeddings as personalized matching scores for news ranking. These personalized news recommendation methods usually model user interests from previous news click behaviors. However, it is difficult for these methods to make accurate recommendation to cold-start users whose behaviors are very sparse (Trevisiol et al., 2014). These users are very common in online news platforms, making the cold-start problem become a critical issue in real systems (Sedhain et al., 2014). Although some methods were proposed to alleviate the cold-start problem in personalized recommendation (Sedhain et al., 2014; Trevisiol et al., 2014), they usually utilized side information (Son, 2016) such as social network (Lin et al., 2014) to enhance user interest modeling. However, the side information used in these methods may be unavailable in news recommendation. In addition, these personalized methods tend to recommend similar news with those users have already read, which makes it difficult for users to receive new news information and may hurt their news reading experience (Nguyen et al., 2014; Wu et al., 2019f). Different from these methods, in *PP-Rec* we consider not only users' personal interest in news but also the popularity of candidate news, which can alleviate both cold-start and diversity problems to some extent.

2.2 Popularity-based News Recommendation

Our work is also related to popularity-based news recommendation methods. Different from personalized news recommendation methods which rank candidate news based on users' personal interests, popularity-based news recommendation methods rank candidate news based on their popularity (Phelan et al., 2009; Tatar et al., 2014; Lerman and Hogg, 2010; Szabo and Huberman, 2010; Jonnalagedda et al., 2016). A core problem in popularity-based news recommendation methods is how to estimate the popularity of candidate news accurately. Most existing methods estimated news popularity based on the statistics of users' interactions with news on online news platforms, such as the number of views and comments (Yang, 2016; Tatar et al., 2014; Lee et al., 2010). For example, Yang (2016) proposed to use the frequency of views to measure news popularity. Tatar et al. (2014) proposed to predict news popularity based
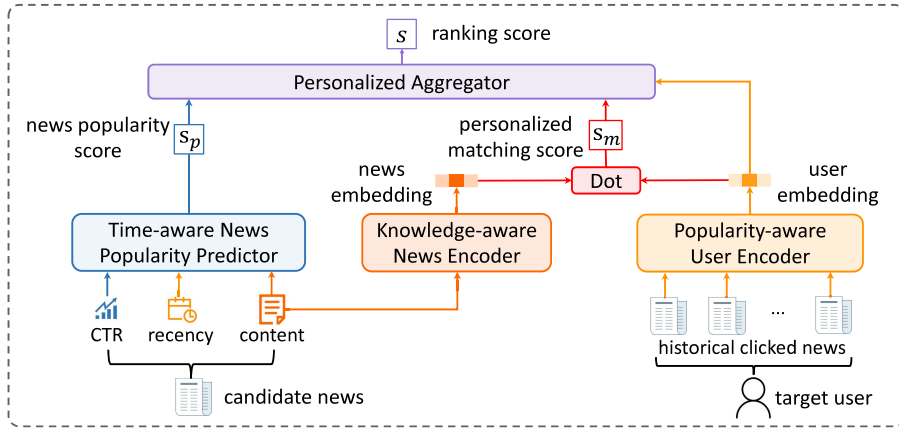

Figure 2: The overall framework of *PP-Rec*.

on the number of comments of news via a linear model. Li et al. (2011) proposed to use the number of clicks on news to model their popularity and further adjust the ranking of news with same topics based on their popularity. However, different news usually have significant differences in impression opportunities, and these view and comment numbers are biased by impression times. Different from these methods, we use CTR to model news popularity, which can eliminate the impression bias. Besides CTR, we also incorporate the content and recency information of candidate news to predict the popularity of candidate news in a more comprehensive and time-aware manner.

3 Methodology

In this section, we introduce *PP-Rec* for news recommendation which can consider both the personal interest of users and the popularity of candidate news. First, we introduce the overall framework of *PP-Rec*, as shown in Fig. 2. Then we introduce the details of each module in *PP-Rec*, which are shown in Figs. 3, 4 and 5.

3.1 Framework of PP-Rec

In *PP-Rec*, the ranking score of recommending a candidate news to a target user is the combination of a personalized matching score s_m and a news popularity score s_p. The personalized matching score is used to measure the user's personal interest in the content of candidate news, and is predicted based on the relevance between news content embedding and user interest embedding. The news content embedding is generated by a *knowledge-aware news encoder* from both news texts and entities. The user interest embedding is generated by a
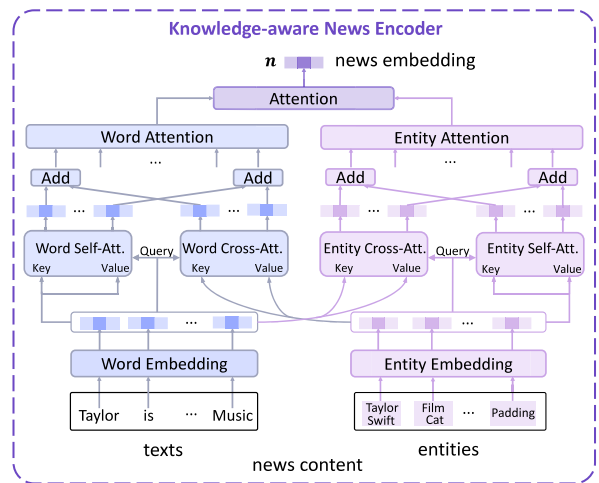

Figure 3: Knowledge-aware news encoder.

popularity-aware user encoder from the content of clicked news as well as their popularity. The news popularity score is used to measure the time-aware popularity of candidate news, which is predicted by a *time-aware news popularity predictor* based on news content, recency, and near real-time CTR.

3.2 Knowledge-aware News Encoder

First, we introduce the *knowledge-aware news encoder*, which is shown in Fig. 3. It learns news representation from both text and entities in news title. Given a news title, we obtain the word embeddings based on word embedding dictionary pre-trained on large-scale corpus to incorporate initial word-level semantic information. We also convert entities into embeddings based on pre-trained entity embeddings to incorporate knowledge information in knowledge graphs to our model.

There usually exists relatedness among entities in the same news. For example, the entity "MAC"

that appears with the entity "Lancome" may indicate cosmetics while it usually indicates computers when appears with the entity "Apple". Thus, we utilize an entity multi-head self-attention network (Vaswani et al., 2017) (MHSA) to learn entity representations by capturing their relatedness. Besides, textual contexts are also informative for learning accurate entity representations. For example, the entity "MAC" usually indicates computers if its textual contexts are "Why do MAC need an ARM CPU?" and indicates cosmetics if its textual contexts are "MAC cosmetics expands AR try-on". Thus, we propose an entity multi-head cross-attention network (MHCA) to learn entity representations from the textual contexts. Then we formulate the unified representation of each entity as the summation of its representations learned by the MHSA and MHCA networks. Similarly, we use a word MHSA network to learn word representations by capturing the relatedness among words and a word MHCA network to capture the relatedness between words and entities. Then we build the unified word representation by adding its representations generated by the word MHSA and the word MHCA networks.

Since different entities usually contribute differently to news representation, we use an entity attention network to learn entity-based news representation $\mathbf{e}$ from entity representations. Similarly, we use a word attention network to learn word-based news representation $\mathbf{w}$ from word representations. Finally, we learn the unified news representation $\mathbf{n}$ with a weighted combination of $\mathbf{e}$ and $\mathbf{w}$ via an attention network.

3.3 Time-aware News Popularity Predictor

Next, we introduce the *time-aware news popularity predictor*, as shown in Fig. 4. It is used to predict time-aware news popularity based on news content, recency, and near real-time CTR information. Since popular news usually have a higher click probability than unpopular news, CTR can provide good clue for popular news (Jiang, 2016). Thus, we incorporate CTR into news popularity prediction. Besides, popularity of a news article usually dynamically changes. Popular news may become less popular as they get out-of-date over time. Thus, we use user interactions in recent t hours to calculate near real-time CTR (denoted as c_t) for news popularity prediction. However, the accurate computation of CTR needs to accumulate
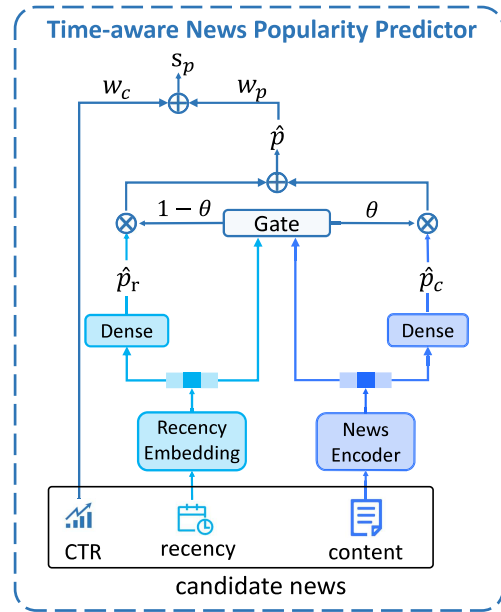

Figure 4: Time-aware news popularity predictor.

sufficient user interactions, which is challenging for those newly published news.

Fortunately, news content is very informative for predicting news popularity. For example, news on breaking events such as earthquakes are usually popular since they contain important information for many of us. Thus, besides near real-time CTR, we incorporate news content into news popularity prediction. We apply a dense network to the news content embedding $\mathbf{n}$ to predict the content-based news popularity $\hat{p}_c$. Since news content is time-independent and cannot capture the dynamic change of news popularity, we incorporate news recency information, which is defined as the duration between the publish time and the prediction time. It can measure the freshness of news articles, which is useful for improving content-based popularity prediction. We quantify the news recency r in hours and use a recency embedding layer to convert the quantified news recency into an embedding vector $\mathbf{r}$. Then we apply a dense network to $\mathbf{r}$ to predict the recency-aware content-based news popularity $\hat{p}_r$. Besides, since different news content usually have different lifecycles, we propose to model time-aware content-based news popularity $\hat{p}$ from $\hat{p}_c$ and $\hat{p}_r$ using a content-specific aggregator:

$$\hat{p} = \theta \cdot \hat{p}_c + (1-\theta) \cdot \hat{p}_r, \ \theta = \sigma(\mathbf{W}^p \cdot [\mathbf{n}, \mathbf{r}] + \mathbf{b}^p), \ (1)$$

where $\theta \in (0, 1)$ means the content-specific gate, $\sigma(\cdot)$ means the sigmoid activation, $[\cdot, \cdot]$ means the
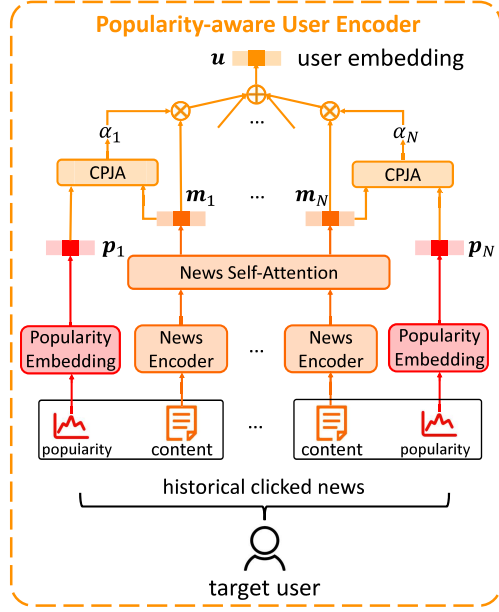
Figure 5: Popularity-aware user encoder.

concatenation operation, $\mathbf{W}^p$ and $\mathbf{b}^p$ are the trainable parameters. Finally, the final time-aware news popularity s_p is formulated as a weighted summation of the content-based popularity $\hat{p}$ and the CTR-based popularity c_t, i.e., $s_p = w_c \cdot c_t + w_p \cdot \hat{p}$, where w_c and w_p are the trainable parameters.

3.4 Popularity-aware User Encoder

Next, we introduce the *popularity-aware user encoder* in *PP-Rec* for user interest modeling, which is shown in Fig. 5. In general, news popularity can influence users' click behaviors, and causes bias in behavior based user interest modeling (Zheng et al., 2010). Eliminating the popularity bias in user behaviors can help more user interest from user behaviors more accurately. For example, a user may click the news "Justin Timberlake unveils the song" because he likes the songs of "Justin Timberlake", while he may click the news "House of Representatives impeaches President Trump" because it is popular and contains breaking information. Among these two behaviors, the former is more informative for modeling the user interest. Thus, we design a *popularity-aware user encoder* to learn user interest representation from both content and popularity of clicked news. It contains three components, which we will introduce in details.

First, motivated by Wu et al. (2019e), we apply a news multi-head self-attention network to the representations of clicked news to capture their relatedness and learn contextual news representation. Second, we uniformly quantify the popularity of the

i-th clicked news predicted by the *time-aware news popularity predictor*[2] and convert it into an embedding vector $\mathbf{p}_i$ via popularity embedding. Third, besides news popularity, news content is also useful for selecting informative news to model user interest (Wu et al., 2019a). Thus, we propose a content-popularity joint attention network (CPJA) to alleviate popularity bias and select important clicked news for user interest modeling, which is formulated as:

$$\alpha_i = \frac{\exp(\mathbf{q}^T \cdot \tanh(\mathbf{W}^u \cdot [\mathbf{m}_i, \mathbf{p}_i]))}{\sum_{j=1}^{N} \exp(\mathbf{q}^T \cdot \tanh(\mathbf{W}^u \cdot [\mathbf{m}_j, \mathbf{p}_j]))}, \quad (2)$$

where α_i and $\mathbf{m}_i$ denote the attention weight and the contextual news representation of the i-th clicked news respectively. $\mathbf{q}$ and $\mathbf{W}^u$ are the trainable parameters. The final user interest embedding $\mathbf{u}$ is formulated as a weighed summation of the contextual news representations: $\mathbf{u} = \sum_{i=1}^{N} \alpha_i \cdot \mathbf{m}_i$.

3.5 News Ranking and Model Training

In this section, we introduce how we rank the candidate news and train the model in detail. The ranking score of a candidate news for a target user is based on the combination of a personalized matching score s_m and a news popularity score s_p. The former is computed based on the relevance between user embedding $\mathbf{u}$ and news embedding $\mathbf{n}$. Following Okura et al. (2017), we adopt dot product to compute the relevance. The latter is predicted by the *time-aware news popularity predictor*. In addition, the relative importance of the personalized matching score and the news popularity score is usually different for different users. For example, the news popularity score is more important than the personalized matching score for cold-start users since the latter is derived from scarce behaviors and is usually inaccurate. Thus, we propose a personalized aggregator to combine the personalized matching score and news popularity score:

$$s = (1 - \eta) \cdot s_m + \eta \cdot s_p, \quad (3)$$

where s denotes the ranking score, and the gate η is computed based on the user representation $\mathbf{u}$ via a dense network with sigmoid activation.

We use the BPR pairwise loss (Rendle et al., 2009) for model training. In addition, we adopt the negative sampling technique to select a negative

---

[2]We remove news recency and content here to avoid non-differentiable quantization operation.

sample for each positive sample from the same impression. The loss function is formulated as:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \log(\sigma(s_i^p - s_i^n)), \qquad (4)$$

where s_i^p and s_i^n denote the ranking scores of the i-th positive and negative sample respectively, and $\mathcal{D}$ denotes the training dataset.

4 Experiment

4.1 Dataset and Experimental Settings

To our best knowledge, there is no off-the-shelf news recommendation dataset with news popularity information. Thus, we built two datasets by ourselves. The first one is collected from the user logs in the Microsoft News website from October 19 to November 15, 2019, and is denoted as *MSN*. We use the user logs in the last week for evaluation and others for model training and validation. The second dataset is collected from a commercial news feeds in Microsoft from January 23 to April 23, 2020, and is denoted as *Feeds*. We use the logs in the last three weeks for evaluation and the rest for model training and validation. For both datasets, we randomly sample 500k impressions for model training, 100k impressions for validation, and 500k impressions for evaluation, respectively. The detailed statistics are listed in Table 1. Following previous works (Wu et al., 2019a; An et al., 2019), we use AUC, MRR, nDCG@5, and nDCG@10 to evaluate recommendation performance.

	# News	# Users	# Impressions	# Clicks
MSN	161,013	490,522	1,100,000	1,675,084
Feeds	4,117,562	98,866	1,100,000	2,384,976

Table 1: Statistics of the datasets.

In our experiments, word embeddings are 300-dimensional and initialized by the Glove embeddings (Pennington et al., 2014). The entity embeddings are 100-dimensional vectors pre-trained on knowledge tuples extracted from WikiData via TransE (Bordes et al., 2013). We use clicked and unclicked impressions in the recent one hour to compute the near real-time CTR. The recency and popularity embeddings are set to 100 dimensions and initialized randomly. All multi-head attention networks are set to have 20 attention heads and the output dimension of each head is 20. All gate networks are implemented by a two-layer

dense network with 100-dimensional hidden vectors. Dropout approach (Srivastava et al., 2014) is applied to *PP-Rec* to migrate overfitting. The dropout probability is set to 0.2. Adam (Kingma and Ba, 2015) is used for model training with 10^{-4} learning rate. Hyper-parameters of *PP-Rec* and baselines are tuned based on the validation set.

4.2 Performance Evaluation

We compare *PP-Rec* with two groups of baselines. The first group is popularity-based news recommendation methods, including: (1) *ViewNum* (Yang, 2016): using the number of news view to measure news popularity; (2) *RecentPop* (Ji et al., 2020): using the number of news view in recent time to measure news popularity; (3) *SCENE* (Li et al., 2011): using view frequency to measure news popularity and adjusting the ranking of news with same topics based on their popularity; (4) *CTR* (Ji et al., 2020): using news CTR to measure news popularity. The second group is personalized news recommendation methods, containing: (1) *EBNR* (Okura et al., 2017): utilizing an auto-encoder to learn news representations and a GRU network to learn user representations; (2) *DKN* (Wang et al., 2018): utilizing a knowledge-aware CNN network to learn news representations from news titles and entities; (3) *NAML* (Wu et al., 2019a): utilizing attention network to learn news representations from news title, body and category; (4) *NPA* (Wu et al., 2019b): utilizing personalized attention networks to learn news and user representations; (5) *NRMS* (Wu et al., 2019e): utilizing multi-head self-attention networks to learn both news and user representations; (6) *LSTUR* (An et al., 2019): modeling users' short-term interests via the GRU network and long-term interests via the user ID; (7) *KRED* (Liu et al., 2020): learning news representation from titles and entities via a knowledge graph attention network.

We repeat each experiment 5 times and show average performance and standard deviation in Table 2, from which we have the following observations. First, among the popularity-based news recommendation methods, the *CTR* method outperforms the *ViewNum* method. This is because the number of news views is influenced by impression bias while CTR can eliminate the impression bias and better measure news popularity. Second, *PP-Rec* outperforms all popularity-based methods. This is because these methods usually recommend popular news to different users. However, differ-

Methods	MSN				Feeds			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
ViewNum	54.12±0.00	24.95±0.00	26.07±0.00	31.56±0.00	58.99±0.00	23.71±0.00	26.83±0.00	32.38±0.00
RecentPop	55.67±0.00	28.72±0.00	30.45±0.00	36.62±0.00	56.27±0.00	24.93±0.00	28.37±0.00	33.89±0.00
SCENE	57.89±0.02	27.41±0.01	28.81±0.02	34.36±0.03	60.82±0.03	27.29±0.03	31.25±0.02	36.56±0.03
CTR	65.72±0.00	30.50±0.00	32.79±0.00	38.68±0.00	66.40±0.00	30.29±0.00	35.53±0.00	40.72±0.00
EBNR	63.90±0.20	30.13±0.12	32.25±0.14	38.05±0.14	64.88±0.04	28.91±0.03	33.29±0.03	38.87±0.02
DKN	64.16±0.19	30.63±0.10	32.98±0.12	38.66±0.11	66.30±0.11	30.25±0.06	35.01±0.07	40.55±0.06
NAML	66.06±0.17	32.10±0.10	34.73±0.11	40.43±0.11	67.50±0.09	31.07±0.08	36.08±0.10	41.61±0.10
NPA	65.83±0.20	31.70±0.09	34.24±0.10	39.96±0.10	67.25±0.10	30.80±0.05	35.72±0.07	41.25±0.07
NRMS	66.34±0.16	32.00±0.08	34.68±0.09	40.39±0.09	68.10±0.05	31.47±0.03	36.61±0.03	42.12±0.03
LSTUR	66.69±0.16	32.12±0.05	34.76±0.05	40.51±0.04	67.43±0.16	30.95±0.11	35.92±0.16	41.45±0.14
KRED	66.54±0.17	31.97±0.14	34.65±0.14	40.38±0.14	67.67±0.18	31.16±0.13	36.19±0.16	41.72±0.16
PP-Rec	**71.05**±0.09	**39.34**±0.08	**44.01**±0.13	**50.46**±0.20	**72.11**±0.21	**32.42**±0.12	**38.13**±0.08	**43.50**±0.13

Table 2: News recommendation results of different methods. We perform t-test and the results show that *PP-Rec* significantly outperforms other baseline methods at significance level $p < 0.001$.

ent users might prefer different news according to their personalized interests, some of which are not popular and cannot be recommended by these popularity-based methods. In contrast, *HieRec* considers both popularity and personalization in news recommendation. Third, *PP-Rec* outperforms all personalized methods. This is because personalized methods usually recommend news based on the matching between news and user interest inferred from users' clicked news, and they ignore the popularity of each news. However, popular news usually contain important and eye-catching information and can attract the attention of many users with different interests. Different from these personalized methods, *PP-Rec* incorporates news popularity into personalized news recommendation, which can recommend popular news to users and improve the performance of news recommendation.

4.3 Performance on Cold-Start Users

We evaluate the performance of *PP-Rec* and several personalized methods on news recommendation for cold-start users. We compare *PP-Rec* with *NAML*, *KRED*, *LSTUR* and *NMRS* since they achieve good performance in Table 2. We evaluate their performance on recommending news to users with $K \in \{k|k = 0, 1, 3, 5\}$ historical clicked news. In the following sections, we only show experimental results on the *MSN* dataset since results on *MSN* dataset and *Feeds* dataset are similar. As shown in Fig. 6, *PP-Rec* significantly outperforms other personalized methods. This is because these personalized methods usually recommend news based on the matching between news and user interests. However, it is difficult for these methods to accurately model personal interests of cold-start users from their scarce clicks and accurately help them
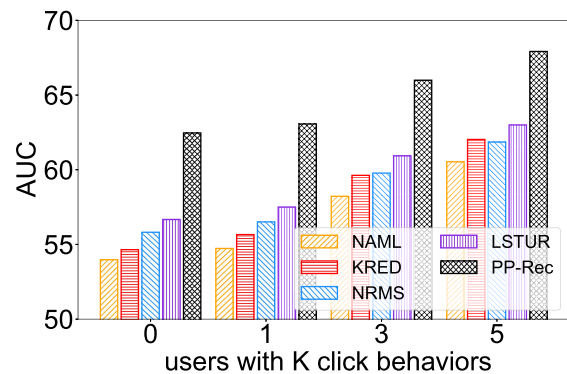

Figure 6: Performance on cold-start users.

find their interested news. Different from these methods, *PP-Rec* recommends news based on both personalized interest matching and news popularity. Popular news usually contains important information and can attract many users with different interests. Thus, incorporating news popularity into news recommendation can effectively improve the reading experiences of cold-start users.

4.4 Recommendation Diversity

In this section, we evaluate the recommendation diversity of *PP-Rec* and other personalized methods. We use two metrics, i.e., intra-list average distance and new topic ratio, to measure the diversity of the top K ($K \in \{k|k = 1, ..., 10\}$) recommended news. The former is used to measure the average distance between recommended news based on their representations, which is widely used in previous works (Zhang and Hurley, 2008; Chen et al., 2018). The second one is used to measure the topic similarity between recommended news and users' historical clicked news. It counts the number of topics of the top K recommended news
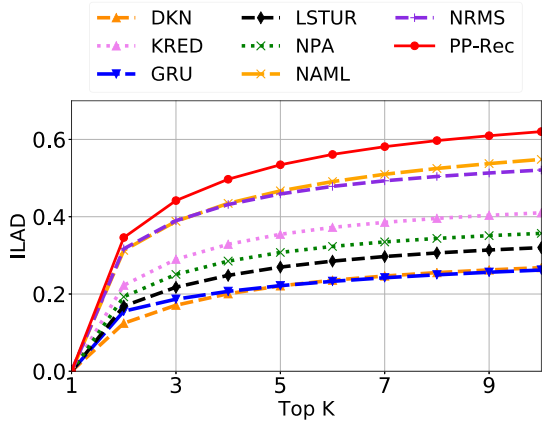
Figure 7: Intra-list average distance of news recommended by different methods.
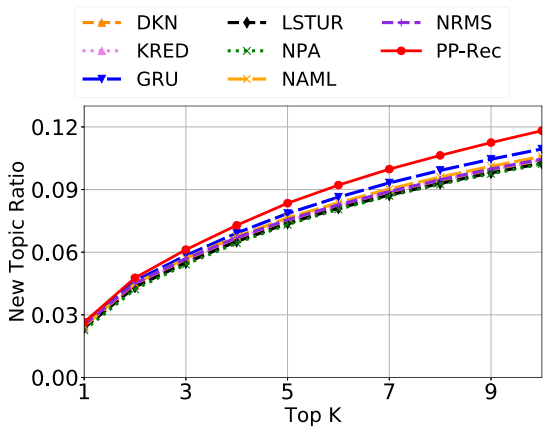

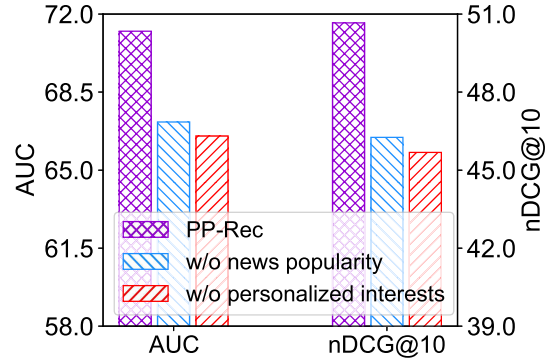Figure 8: New topic ratio of news recommended by different methods.

Figure 9: Effectiveness of personalized matching score and news popularity score.
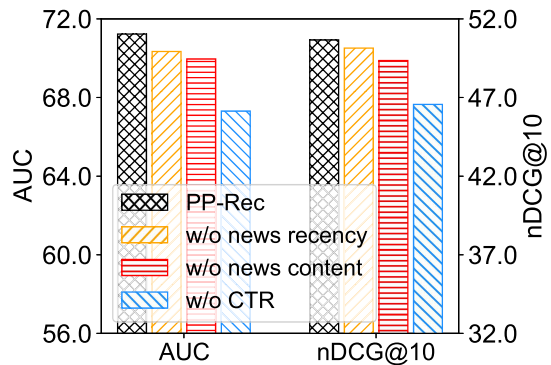
Figure 10: Effectiveness of different information used for news popularity prediction.

which are clicked and are not included in topics of users' historical clicked news. Besides, we use K to normalize the number. Fig. 7 and 8 show that *PP-Rec* can consistently improve the recommendation diversity. This is because these personalized methods recommend news to users based on the matching between news and user interest inferred from clicked news, making the recommended news tend to be similar to users' consumed news. Different from these methods, *PP-Rec* incorporates news popularity into news recommendation. Besides the news which is related to user interest, *PP-Rec* can also recommend popular news, which are very diverse in content and topics, to users. Thus, *PP-Rec* can enhance recommendation diversity.

4.5 Ablation Study

In this section, we conduct several ablation studies on *PP-Rec*. First, we verify the effectiveness of the two scores for candidate news ranking, i.e., news popularity score and personalized matching score,

by removing them individually from *PP-Rec*. The experimental results are shown in Fig. 9. We have two findings from the results. First, after removing the news popularity score, the performance of *PP-Rec* declines. This is because *PP-Rec* incorporates news popularity into news recommendation via this score. In addition, popular news usually contains important information and can attract many users with different interests. Thus, recommending popular news can improve news recommendation accuracy. Second, removing the personalized matching score also hurts the recommendation accuracy. This is because this score measures user interest in news and incorporates personalized matching into news recommendation in *PP-Rec*. Since users like to click news related to their personalized interests, recommending users' interested news can effectively improve recommendation accuracy.

Next, as shown in Fig. 10, we conduct an ablation study to verify the effectiveness of different information in the *time-aware news popularity predictor* by removing them individually. We have several observations from the results. First, removing news recency makes the performance of *PP-Rec*

Historical clicked news of the user	Top recommendations from PP-Rec		Top recommendations from LSTUR	
Title	Title	Popularity	Title	Popularity
Frustrated Antonio Brown has active morning on twitter.	**For grandfather charged in girl's cruise ship death, video could be key.**	0.156	**Jared Goff regression: Here's exactly what's gone wrong for rams.**	0.031
Tom Brady: When it comes to the future, my focus is on this season.	Patriots-Ravens part II seems inevitable.	0.016	Bill Belichick irritated with questions about Antonio brown.	0.036
Odell Beckham Jr. trolled Steelers coach Mike Tomlin with yawn celebration.	**Jared Goff regression: Here's exactly what's gone wrong for rams.**	0.031	Patriots-Ravens part II seems inevitable.	0.016

Figure 11: Top news recommended by *PP-Rec* and *LSTUR*. The clicked news are in red and bold.

decline. This is because news popularity usually dynamically changes, and popular news will become unpopular once its information is expired. Since news recency can reflect the freshness of news information, incorporating it makes the news popularity modeling more accurate. Second, the performance of *PP-Rec* without news content also declines. This is because after removing it, *PP-Rec* predicts news popularity based on the near real-time CTR and recency. However, it usually takes some time to accumulate enough impressions to calculate accurate CTR. Thus, removing the news content makes *PP-Rec* cannot effectively model the popularity of news just published. Third, *PP-Rec* performs worse without the near real-time CTR. This is because near real-time CTR effectively measures the click probability of the news based on the behaviors of a large number of users in the recent period. Thus, removing the near real-time CTR makes it *PP-Rec* lose much useful information for modeling the dynamic news popularity.

4.6 Case Study

We conduct a case study to show the effectiveness of *PP-Rec*. We compare *PP-Rec* with *LSTUR* since *LSTUR* can achieve the best performance among baseline methods on the *MSN* dataset. In Fig. 11, we list top 3 news recommended by two methods to a randomly sampled user and their normalized popularity predicted by *PP-Rec*. We also list user's clicked news. First, we find that the user clicked a news on football, which is recommended by both *LSTUR* and *PP-Rec*. This is because the user has previously clicked three news on football, which indicates the user is interested in football. Thus, both *LSTUR* and *PP-Rec* recommend that news based on the personal interest of this user. Second, the user did not click other news on football recommended by *PP-Rec* and *LSTUR*. This may be because recommending too much news with similar information may make users feel bored, making the user only click a part of them. This inspires us

that recommending news with diverse information may help improve users' reading experience. Third, the user clicked a news on crime, which is only recommended by *PP-Rec*. This is because it is hard to predict user's interests in criminal events from her clicks, making it difficult for *LSTUR* to recommend this news. Different from *LSTUR*, *PP-Rec* recommends news based on both personal user interest and news popularity. *PP-Rec* successfully predicts that this news is popular and recommends it. This case shows that *PP-Rec* can improve the recommendation accuracy and enhance the recommendation diversity by incorporating news popularity.

5 Conclusion

In this paper, we propose a new news recommendation method named *PP-Rec* to alleviate the cold-start and diversity problems of personalized news recommendation, which can consider both the personal interest of users and the popularity of candidate news. In our method, we rank the candidate news based on the combination of a personalized matching score and a news popularity score. We propose a unified model to predict time-aware news popularity based on news content, recency, and near real-time CTR. In addition, we propose a knowledge-aware news encoder to generate news content embeddings from news texts and entities, and a popularity-aware user encoder to generate user interest embeddings from the content and popularity of clicked news. Extensive experiments on two real-world datasets constructed by logs of commercial news websites and feeds in Microsoft validate that our method can effectively improve the accuracy and diversity of news recommendation.

Acknowledgments

References

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *ACL*, pages 336–345.

Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *RecSys.*, pages 195–202.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.

Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *NIPS*, pages 5622–5633.

Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *WWW*, pages 2863–2869.

Zied Ben Houidi, Giuseppe Scavo, Stefano Traverso, Renata Teixeira, Marco Mellia, and Soumen Ganguly. 2019. The news we like are not the news we visit: news categories popularity in usage data. In *ICWSM*, pages 91–102.

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A re-visit of the popularity baseline in recommender systems. In *SIGIR*, pages 1749–1752.

Zilong Jiang. 2016. Research on ctr prediction for contextual advertising based on deep architecture model. *CEAI*, pages 11–19.

Nirmal Jonnalagedda, Susan Gauch, Kevin Labille, and Sultan Alfarhood. 2016. Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science*, page 63.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Jong Gun Lee, Sue Moon, and Kave Salamatian. 2010. An approach to model and predict the popularity of online contents with explanatory factors. In *WI-IAT*, pages 623–630.

Kristina Lerman and Tad Hogg. 2010. Using a model of social dynamics to predict popularity of news. In *WWW*, pages 621–630.

Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. Scene: a scalable two-stage personalized news recommendation system. In *SIGIR*, pages 125–134.

Chen Lin, Runquan Xie, Xinjun Guan, Lei Li, and Tao Li. 2014. Personalized news recommendation via implicit social experts. *Information Sciences*, pages 1–18.

Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. Kred: Knowledge-aware document representation for news recommendations. In *RecSys.*, pages 200–209.

Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*, pages 31–40.

Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *WWW*, pages 677–686.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Owen Phelan, Kevin McCarthy, and Barry Smyth. 2009. Using twitter to recommend real-time topical news. In *RecSys.*, pages 385–388.

Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning. In *EMNLP: Findings*, pages 1423–1432.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461.

Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. In *RecSys.*, pages 345–348.

Le Hoang Son. 2016. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*, 58:87–104.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, pages 1929–1958.

Gabor Szabo and Bernardo A Huberman. 2010. Predicting the popularity of online content. *Communications of the ACM*, pages 80–88.

Alexandru Tatar, Panayotis Antoniadis, Marcelo Dias De Amorim, and Serge Fdida. 2014. From popularity prediction to ranking online news. *SNAM*, pages 174–186.

Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. 2014. Cold-start news recommendation with domain-dependent browse graph. In *RecSys.*, pages 81–88.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *WWW*, pages 1835–1844.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. *IJCAI*, pages 3863–3869.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. Npa: Neural news recommendation with personalized attention. In *KDD*, pages 2576–2584.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019c. Neural news recommendation with topic-aware news representation. In *ACL*, pages 1154–1159.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019d. Neural news recommendation with heterogeneous user behavior. In *EMNLP*, pages 4876–4885.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019e. Neural news recommendation with multi-head self-attention. In *EMNLP*, pages 6390–6395.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020a. Sentirec: Sentiment diversity-aware neural news recommendation. In *AACL*, pages 44–53.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020b. User modeling with click preference and reading satisfaction for news recommendation. In *IJCAI*, pages 3023–3029.

Chuhan Wu, Fangzhao Wu, Tao Qi, Jianxun Lian, Yongfeng Huang, and Xing Xie. 2020c. Ptum: Pre-training user model from unlabeled user behaviors via self-supervision. In *EMNLP: Findings*, pages 1939–1944.

Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. Fairrec:fairness-aware news recommendation with decomposed adversarial learning. In *AAAI*.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020d. Mind: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606.

Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019f. Pd-gan: Adversarial learning for personalized diversity-promoting recommendation. In *IJCAI*, pages 3870–3876.

JungAe Yang. 2016. Effects of popularity-based news recommendations ("most-viewed") on users' exposure to online news. *Media Psychology*, pages 243–271.

Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys.*, pages 123–130.

Hua Zheng, Dong Wang, Qi Zhang, Hang Li, and Tinghao Yang. 2010. Do clicks measure recommendation relevancy? an empirical user study. In *RecSys.*, pages 249–252.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Guo Li. 2019. Dan: Deep attention neural network for news recommendation. In *AAAI*, pages 5973–5980.