Accurate News Recommendation Coalescing Personal and Global Temporal Preferences

Bonhun Koo, Hyunsik Jeon, and U Kang(✉)

Seoul National University, Seoul, South Korea
{darkgs,jeon185,ukang}@snu.ac.kr

Abstract. Given session-based news watch history of users, how can we precisely recommend news articles? Unlike other items for recommendation, the worth of news articles decays quickly and various news sources publish fresh ones every second. Moreover, people frequently select news articles regardless of their personal preferences to understand popular topics at a specific time. Conventional recommendation methods, designed for other recommendation domains, give low performance because of these peculiarities of news articles.

In this paper, we propose PGT (News Recommendation Coalescing **P**ersonal and **G**lobal **T**emporal Preferences), an accurate news recommendation method designed with consideration of the above characteristics of news articles. PGT extracts latent features from both personal and global temporal preferences to sufficiently reflect users' behaviors. Furthermore, we propose an attention based architecture to extract adequate coalesced features from both of the preferences. Experimental results show that PGT provides the most accurate news recommendation, giving the state-of-the-art accuracy.

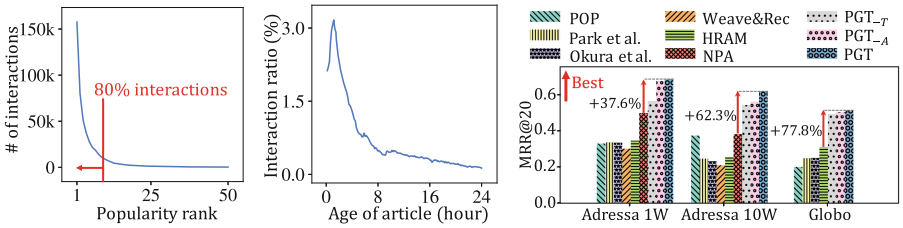
Keywords: News recommender systems · Personal and global temporal preferences · Attention · Recurrent neural network

1 Introduction

Given news articles and watch history of users, how can we accurately recommend news articles to users? Even though online news service has become a main source of news, a massive amount of news articles released everyday makes it difficult for users to search for articles of their interests. Thus, it is crucial for online news providers to recommend appropriate news articles for users to improve their experiences.

In online news services that provide news to customers, consumptions are extremely skewed to spotlighted news. Figure 1a shows the skewness of consumptions in Adressa dataset (see Sect. 4.1 for details), a real-world news service dataset; x-axis indicates the popularity ranks of news which could vary over time, and y-axis indicates the number of consumptions. As shown in the figure, interactions between users and news have *popularity pattern*, meaning that users

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 78–90, 2020.
https://doi.org/10.1007/978-3-030-47426-3_7



(a) Popularity pattern (b) Freshness pattern (c) Accuracy of methods

Fig. 1. (a) *popularity pattern*: users tend to prefer *popular* news. (b) *freshness pattern*: users prefer relatively *fresh* news. (c) Our proposed PGT gives the best accuracy, thanks to its consideration of both global and personal preferences.

mostly prefer popular news. For instance, the top-7 most popular articles account for 80% of total consumptions during the entire time. Figure 1b shows a *freshness pattern* in Adressa dataset, where the number of consumptions of each news rapidly decreases over its age. This indicates that customers prefer relatively fresh news since the novelty of news expires quickly over time. Thus, the challenge of designing an accurate news recommender systems is to consider the global temporal preference while taking into account each user's personal preference.

In this paper, we propose PGT (News Recommendation Coalescing **P**ersonal and **G**lobal **T**emporal Preferences), a novel approach for news recommendation considering both of personal and global temporal preferences. PGT gives a recommendation for each user at time t leveraging 1) the global temporal preference at time t, and 2) watch-history of the user before t.

1. **Global temporal preference.** Representing global temporal preference at time t as a low-rank latent vector is challenging since it has to involve both of popularity pattern and freshness pattern at that time. To deal with this challenge, PGT selects 1) document vectors of the most popular n articles at time t, which stand for popularity pattern, and 2) document vectors of the most fresh m articles at time t, which stand for freshness pattern. These vectors for the two different patterns are combined by the self-attention [15] with a scaled dot product (details in Sect. 3.2).
2. **Watch history.** A user's previous watch history epitomizes the user's personal preference. The goal of PGT is to extract a sequential pattern in the watch history with regard to the global temporal preference. For the purpose, PGT uses a bidirectional LSTM (BiLSTM) which has shown the best performance in encoding session-based sequential data [5] where a session is a set of consecutive behaviors of a user. PGT combines the hidden states of BiLSTM by an attention network using the global temporal preference as context (details in Sect. 3.3).

PGT then generates a prediction vector using a fully-connected neural network where information from both of personal preference and global temporal

preference at time t is fed into. Lastly, PGT ranks candidate articles based on the similarity between the embedding vectors of the candidates and the generated prediction vector.

We summarize our main contributions as follows:

- **Modeling personal and global temporal preference.** We introduce *global temporal preference* which indicates the comprehensive pattern of all users in news services. PGT models how global temporal preference influences each user's personal preference.
- **Attention-based architecture coalescing preferences.** We propose an attention-based network architecture to dynamically control weights of features in 1) the representation of global temporal preference, and 2) the representation of personal preference. The global temporal preference vector is used as context in the attention network for the personal preference. Attention helps PGT effectively deal with a quick change of personal preference in the online news ecosystem.
- **Experiment.** Extensive experimental results show that PGT provides the best accuracy, outperforming competitors by significant margins (see Fig. 1c).

In the rest of paper, we review related works in Sect. 2, introduce our proposed method PGT in Sect. 3, evaluate PGT and competitors in Sect. 4, and conclude in Sect. 5.

2 Related Works

We review previous researches on news recommendation systems.

Early studies on recommendation systems use variants of recurrent neural network (RNN) to model input sequences [2,3]. However, these methods usually suffer from the cold-start problems. To deal with the cold-start problems, several studies enrich the embedding of newly published articles by utilizing the meta-information of articles [4]. Okura et al. [11] proposed a news recommender system based on variational autoencoder (VAE) and RNN. They used a method based on VAE to learn embedding vectors of articles, such that vectors in the same categories are made similar. After learning article embeddings, they used RNNs to predict the next article vector that a user is likely to watch. Park et al. [12] proposed a news recommendation system based on RNN to model each user's personal preference. They reranked the candidates by each user's long-term categorical preference which is the weighted sum of categories of news articles that the user has seen; this categorical preference improved the accuracy.

Recent studies proposed attention-based methods to model users' behaviors without RNNs. Wang et al. [16] proposed a deep news recommendation method with a CNN model [7] and a news-level attention network. They enhanced their method using the embeddings of the entities extracted from a knowledge graph. Chuhan et al. [17] proposed to use attention networks at both word-level and news-level to highlight informative words and news. They also proposed personalized attention by using the embedding of user ID as context to differentially attend to important words and news according to personal preference.

We note that none of the above methods consider the global temporal preference, and its coalesced relation to personal preference. Thus they show poor performance compared to our proposed PGT (see Sect. 4).

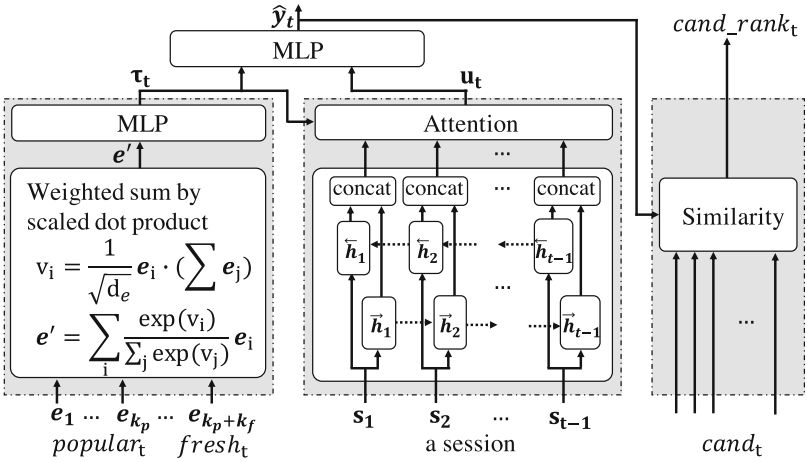


Fig. 2. Architecture of PGT. To recommend news articles to a user at time t, PGT generates a prediction vector $\hat{\boldsymbol{y}}_t$ using previous article $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{t-1}$ in the session of the user, popular articles $popular_t$ at time t, and fresh articles $fresh_t$ at time t. PGT ranks candidate articles $cand_t$ based on their similarities to the prediction vector $\hat{\boldsymbol{y}}_t$.

3 Proposed Method

We propose PGT, an accurate method for news recommendation. We first provide a brief overview of our method in Sect. 3.1. Then we describe how to generate 1) a representation for the global temporal preference in Sect. 3.2, and 2) a representation for the personal preference in Sect. 3.3. Finally, we introduce how to rank the candidate articles in Sect. 3.4.

3.1 Overview

We design PGT for balanced extraction of latent features from both of personal preference and global temporal preference. PGT reflects users' online news watch behaviors by accumulating these two preferences to provide an accurate recommendation. We concentrate on the following challenges to address in a news recommendation system.

1. **Cold-Start problem.** News recommendation systems need to recommend newly published articles which have no explicit user feedback. How can we provide accurate news recommendation for these fresh articles?

2. **Popularity and freshness patterns.** As shown in Fig. 1, the novelty of news expires quickly. How can we dynamically capture the personal and global temporal preferences to promptly trace changing trend of news?

We address the aforementioned challenges by the following ideas:

1. **Global temporal preference.** PGT extracts the global temporal preference from popular and fresh articles at recommendation time. The global temporal preference represents the time-dependent features and helps PGT consider newly published articles well (details in Sect. 3.2).
2. **RNN-attention using global temporal preference as context.** PGT embeds each user's personal preference from its watch history using an RNN with an attention network. The attention network determines the importance of each previous user behavior. We use the global temporal preference as context of the attention network to consider both popularity and freshness patterns (details in Sect. 3.3).

Figure 2 shows the overall architecture of PGT. We use Doc2Vec [10] to get embedding vectors for popular articles $popular_t$, fresh articles $fresh_t$, articles in a *session*, and candidate articles $cand_t$ at time t. PGT generates a prediction vector $\hat{\boldsymbol{y}}_t$ from personal preference $\boldsymbol{u_t}$ and global temporal preference $\boldsymbol{\tau_t}$. Finally, PGT ranks candidate articles $cand_t$ by their similarities to $\hat{\boldsymbol{y}}_t$.

3.2 Global Temporal Preference

Our goal is to generate a single vector of global temporal preference from several popular and fresh articles. The global temporal preference vector represents common interests shared by all users at a specific time. For the purpose, we adapt self-attention [15]; however, instead of generating vector representation for each key, we generate the single vector by computing the weighted average of article vectors using the attention weights. PGT generates the global temporal preference vector $\boldsymbol{\tau}_t$ as follows:

$$v_{g_i} = \frac{1}{\sqrt{d_e}} \boldsymbol{e}_i^T \sum_{\boldsymbol{e}_j \in E} \boldsymbol{e}_j$$

$$\boldsymbol{\tau}_t = W_{g_a} \sum_{\boldsymbol{e}_i \in E} \frac{exp(v_{g_i})}{\sum_j exp(v_{g_j})} \boldsymbol{e}_i + \boldsymbol{b}_{g_a}$$

where a linear layer, consisting of parameter W_{g_a} and $\boldsymbol{b}_{g_a}$, calculates the global temporal preference vector $\boldsymbol{\tau}_t \in \mathbb{R}^n$ at time t. The weighted average of popular and fresh article vectors $\boldsymbol{e}_i \in E = \{popular_t \cup fresh_t\}$ passes through the linear layer where the weights come from the unnormalized attention score v_{g_i} for each article i.

3.3 Personal Preference

Considering individual user's personal preference is essential to make personalized news recommendations. We observe that each previous behavior of a user corresponds to a preference at that time. From this observation, we extract each user's personal preference from its previous watch history using an RNN model and aggregate them by an attention network to generate a vector representing the user's news preference. We use bi-directional LSTM (BiLSTM) in the RNN model since it is well known to summarize both the preceding and the following behaviors. Note that the hidden state $\boldsymbol{h}_i$ of BiLSTM represents the behavior of the user at time i. Then we aggregate all previous hidden states $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{t-1}$ with an attention network by utilizing the global temporal preference vector $\boldsymbol{\tau}_t$ as context vector. The attention network traces the popularity and freshness patterns of online news services to highlight important previous hidden states.

PGT generates a personal preference vector $\boldsymbol{u}_t$ as follows:

$$\begin{aligned} \boldsymbol{h}_i &= f(\boldsymbol{s}_i; \theta) \\ v_{u_i} &= W_{u_a}[\boldsymbol{\tau}_t, \boldsymbol{h}_i] + b_{u_a} \\ \boldsymbol{u}_t &= \sum_i \frac{v_{u_i}}{\sum_j v_{u_j}} \boldsymbol{h}_i \end{aligned}$$

where f is a BiLSTM function, $\boldsymbol{h}_i$ is i-th hidden state vector in the session, and $\boldsymbol{s}_i \in \mathbb{R}^{d_e}$ denotes the representation of the i-th selected article in the session. A linear layer, consisting of parameters W_{u_a} and b_{u_a}, takes a concatenated vector $[\boldsymbol{\tau}_t, \boldsymbol{h}_i]$ of $\boldsymbol{\tau}_t$ and $\boldsymbol{h}_i$ as input, and then calculates the unnormalized attention score v_{u_i} for each hidden state. We generate the personal preference vector $\boldsymbol{u}_t$ at time t by calculating the weighted average of hidden states using attention scores.

3.4 Ranking Candidate Articles

To recommend news articles to a user, we rank candidate articles based on the user's interest at each time step. Ideally, candidate articles should include all of the existing news articles; however, in practice we need to narrow down candidates for scalability of recommender systems. We select candidate articles by removing unpopular articles at each time step. In the training process, we ensure that candidate articles contain the actually selected article at each time step by replacing the most unpopular candidate article with the actually selected article.

Given a session of a user, let $\boldsymbol{s}_t \in \mathbb{R}^{d_e}$ denote the representation of the t-th selected article in the session. $\boldsymbol{\tau}_t \in \mathbb{R}^n$ represents the global temporal preference at the t-th time step in the session. As discussed in Sect. 3.2, PGT generates τ_t utilizing both popular articles and fresh articles; this allows PGT to consider popularity and freshness as important factors for recommendation, which alleviates the cold-start problem. $\boldsymbol{u}_t \in \mathbb{R}^n$ denotes a user's personal preference at the t-th time step in the session, allowing personalized news recommendation.

We generate the prediction vector $\hat{\boldsymbol{y}}_t$ as follows:

$$\hat{\boldsymbol{y}}_t = W_o[\boldsymbol{\tau}_t, \boldsymbol{u}_t] + \boldsymbol{b}_o$$

where the vector created by concatenating $\boldsymbol{\tau}_t$ and $\boldsymbol{u}_t$ is passed through a linear layer consisting of parameters W_o and $\boldsymbol{b}_o$. Then we rank candidate articles at the t-th time step based on the similarity to $\hat{\boldsymbol{y}}_t$.

We train PGT to minimize the L2 distance between the truly selected article vector $\boldsymbol{s}_t$ and the prediction vector $\hat{\boldsymbol{y}}_t$ as follows:

$$\mathcal{L}(\boldsymbol{s}_t, \hat{\boldsymbol{y}}_t) = \|\boldsymbol{s}_t - \hat{\boldsymbol{y}}_t\|_2.$$

We backpropagate gradients calculated from this loss function to the MLP, the RNN, and the attention network in PGT, while fixing the article embeddings. As a result, PGT is trained to reduce the distance between $\hat{\boldsymbol{y}}_t$ and $\boldsymbol{s}_t$, which leads to increasing the distance between $\hat{\boldsymbol{y}}_t$ and vectors of unselected articles.

4 Experiment

We run experiments to answer the following questions.

- **Q1. Accuracy (Sect. 4.2).** How well does PGT recommend news articles?
- **Q2. Effect of modeling global temporal preference (Sect. 4.3).** Does the modeling of global temporal preference help improve the accuracy?
- **Q3. Effect of attention network in personal preference (Sect. 4.4).** How well does the attention network for the personal preference help improve the accuracy?

4.1 Experimental Settings

Dataset. We use Adressa [1] and Globo [14] which are session-based datasets of news watch history (see Table 1). Adressa dataset[1] is generated from the behaviors of users in the Adresseavision, a newspaper media in Norway. The one week version Adressa 1W of the dataset contains news information from 1 to 7 January, 2017. The full version Adressa 10W of the dataset contains news information from 1 January to 31 March, 2017. The dataset contains URLs of all articles, and contents of a subset of the articles; articles with invalid URLs are removed. The second dataset Globo[2] [14] contains news information from a news portal G1 (G1.com) from 1 to 16 October, 2017. Instead of revealing the original news contents, it provides the embedding vector for each news article due to license restrictions.

[1] http://reclab.idi.ntnu.no/dataset.
[2] https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom.

Table 1. Summary of news datasets.

Dataset	# Sessions	# Events	# Articles	Period
Adressa 1W[1]	112,405	487,961	11,069	7 days
Adressa 10W[1]	655,790	8,167,390	43,460	90 days
Globo[2]	296,332	2,994,717	46,577	16 days

Competitor. We compare the performance of our proposed PGT to the following competitors.

- **POP.** This method recommends the most popular items regardless of each user's personal preference.
- **Park et al.** [12]**.** To recommend the next article, this method ranks the candidate articles using a hidden vector generated from RNNs, and then reranks candidates by each user's long-term categorical preference. They also proposed a CNN model to infer missing categories of articles from their contents.
- **Okura et al.** [11]**.** This method recommends articles based on the similarity of articles using dot products of their vector representations. The method generates similar vector representations to articles with similar categories.
- **Weave&Rec** [6]**.** This method utilizes the content of news articles as well as the sequence in which the articles were read by users. They use 3-dimensional CNN to embed both 1) the word embeddings of articles, and 2) the sequence of articles selected by users at the same time.
- **HRAM** [5]**.** This method aggregates outputs of two heterogeneous methods which are 1) user-item matrix factorization to model the interaction between users and items, and 2) attention-based recurrent network to trace the interest of each user.
- **NPA** [17]**.** This method uses personalized attention at both word-level and news-level to highlight informative words and news. The personalized attention uses the embedding of each user as context vector to differentially attend to important words and news according to personal preference.

Vectorized Representation of Article. For Adressa dataset, we train Doc2Vec model [10] with Gensim [13], which has shown a good performance on news recommendation [16]. We use sentences of each article if provided by the dataset, or use crawled sentences using the URL of it otherwise. We set the dimension of embedding vector to 1000, and the size of window to 10. We initialize α of Doc2Vec to 0.025, and decrease α by 0.001 for every 10 epochs. Note that these values are selected since they give the best result. For Globo dataset, we use the provided embedding vector for each article from it.

Evaluation Metrics. We evaluate the accuracy of methods using Hit Rate (HR) and Mean Reciprocal Rank (MRR). Given the probability ranks of truly seen news articles, we calculate HR@5 and MRR@20 as follows:

$$HR@5 = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} |\{r_i | r_i <= 5\}|$$

$$MRR@20 = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} c_i, \qquad c_i = \begin{cases} \frac{1}{r_i}, \text{ if } r_i \leq 20 \\ 0, \text{ otherwise} \end{cases}$$

where $i \in \mathcal{I}$ is an index of an article in the test data, and r_i is the estimated rank of i by a method. HR@5 is the proportion of predictions where the truth is within the top 5 articles with the highest scores. MRR@20 gives a higher score when r_i is more accurate but scores nothing if r_i is above 20.

Model Training. All of the competitors and our method are trained using the same hardware and early-stop policy. We divide our session data into training, validation [9], and test sets with ratio of 8:1:1 based on the user interaction time in a session. When a dataset consists of user interactions from 1 to 10 May, for example, data from 1 to 8 May is used as a training set, data on 9 May as a validation set, and data on the last day as a test set, respectively. This setting is useful to show the effect of the cold-start problem, since several fresh articles in the test set are not included in the training set.

Hyperparameters. We train methods to maximize the similarity between a prediction vector and the corresponding selected article vector for every time step. For PGT, we use the mean squared error (MSE) of the two vectors as a loss function, and Adam optimizer [8] as an optimizer. We use the hyperbolic tangent as a non-linear activation function, but omits it for the last MLP since it gives the best accuracy. For the competitors, we follow their best settings. We use mini-batched inputs of size 512 to feed models during training. When the validation loss keeps increasing for 10 epochs, we early stop training to prevent overfitting. All methods in our experiments early stopped before 200 epochs of training.

Table 2. Our proposed PGT shows the best performance for all of the datasets. This table shows the accuracy of PGT and competitors, measured with the hit rate (HR@5) and the mean reciprocal rank (MRR@20); higher values mean better performances.

Dataset	Metric	POP	Park et al. [12]	Okura et al. [11]	Weave&Rec [6]	HRAM [5]	NPA [17]	PGT
Adressa 1W	HR@5	0.4988	0.4714	0.4569	0.4377	0.5347	0.6512	**0.8668**
	MRR@20	0.3291	0.3361	0.3341	0.3013	0.3452	0.4983	**0.6857**
Adressa 10W	HR@5	0.5672	0.3677	0.3477	0.3007	0.3941	0.5819	**0.7106**
	MRR@20	0.3735	0.2461	0.2320	0.2101	0.2531	0.3818	**0.6197**
Globo	HR@5	0.2845	0.3551	0.3537	–	0.4474	–	**0.5663**
	MRR@20	0.2001	0.2483	0.2500	–	0.3101	–	**0.5116**

4.2 Recommendation Accuracy

Table 2 shows accuracies of PGT and competitors; Weave&Rec and NPA, which require news contents, are not evaluated for Globo since it does not provide news contents. Note that PGT gives the highest accuracy compared to the competitors. We have the following observations from the results.

First, even though POP recommends news articles based only on the general popularity, the popularity pattern of news data (shown in Fig. 1) makes POP a strong baseline showing a good performance. Meanwhile, the performances of the other competitors (Park et al. [12], Okura et al. [11], HRAM, Weave&Rec, and NPA) are less accurate especially in Adressa. It is because they consider only personal preference while users' behaviors in Adressa are more skewed toward the popularity pattern. Note that HR@5 of POP is the probability of watching one of the top 5 most popular articles. HR@5 of POP in Adressa 10W, Adressa 1W, and Globo are 0.5672, 0.4988, and 0.2845, respectively; this shows that users' interactions in Adressa are more skewed to popular articles which are more related to global temporal preference rather than personal preference. On the other hand, our proposed PGT shows the best performance even on Adressa compared to the other competitors by appropriately attending to personal and global temporal preferences.

Table 3. The global temporal preference and the attention network of PGT improve the accuracy of recommendation. PGT outperforms both 1) PGT$_{-T}$, a variant of PGT without the global temporal preference (Sect. 3.2), and 2) PGT$_{-A}$, a variant of PGT without the attention network of BiLSTM (Sect. 3.3).

Dataset	Metric	PGT$_{-T}$	PGT$_{-A}$	PGT
Adressa 1W	HR@5	0.6662	0.8497	**0.8668**
	MRR@20	0.5647	0.6756	**0.6857**
Adressa 10W	HR@5	0.6360	0.6946	**0.7106**
	MRR@20	0.5423	0.5610	**0.6197**
Globo	HR@5	0.5366	0.5562	**0.5663**
	MRR@20	0.4923	0.5035	**0.5116**

Second, the methods modeling each user's watch history by utilizing attention network (PGT, HRAM, and NPA) are more accurate compared to the other competitors (Park et al. [12], Okura et al. [11], and Weave&Rec). The attention network dynamically attends to important previous behaviors, and thus increases recommendation accuracy. Meanwhile, due to the property of RNN, inference in RNN-based methods (Park et al. [12] and Okura et al. [11]) often neglects users' long-term behaviors. Weave&Rec captures a temporal pattern of each user's watch history using 3-dimensional CNN, but provides poor recommendations to fresh users because of their insufficient watch histories.

Finally, HRAM and NPA, which train an embedding for each user, show poor performance compared to our proposed PGT. Note that we divide the dataset into training, validation, and test sets based on user interaction time (see Sect. 4.1), since such setting is more realistic for online news recommendation. However, this makes it very hard for HRAM and NPA to train the embeddings of fresh users well. On the other hand, PGT performs accurate news recommendation even in this case, by utilizing the global temporal preference.

4.3 Effect of Modeling Global Temporal Preference

PGT overcomes the cold-start problem by considering the global temporal preference. In Table 2, we compare the performances of PGT and other neural network based methods on Adressa 1W and Adressa 10W to show how well the global temporal preference helps preserve the accuracy from the cold-start problem, since the other methods neglect the global temporal preference. Note that all neural network based methods suffer from the cold-start problem more severely on Adressa 10W since the time gap between the train and the test set is the longest in it. MRR@20s of Park et al. [12], Okura et al. [11], Weave&Rec, HRAM, and NPA decrease by 26.78%, 30.56%, 30.29%, 26.89%, and 21.42%, respectively, on Adressa 10W compared to those on Adressa 1W; on the other hand, MRR@20 of PGT decreases only by 9.12% on the same setting. This shows that PGT better handles the cold-start problem.

To further validate the effect of the global temporal preference in PGT, we evaluate the performance of PGT$_{-T}$, a variant of PGT, that does not use the global temporal preference, but keeps the attention network of BiLSTM by using the most recent hidden state vector as the context of attention. Columns PGT$_{-T}$ and PGT of Table 3 show that 1) MRR@20 of PGT improves by 17.64%, 12.49%, and 3.77% on Adressa 1W, Adressa 10W, and Globo, respectively, and 2) HR@5 of PGT improves by 23.14%, 10.49%, and 5.24% on Adressa 1W, Adressa 10W, and Globo, respectively, compared to PGT$_{-T}$ which does not use the global temporal preference. This result shows that the global temporal preference helps model popularity and freshness patterns well, leading to a better performance.

4.4 Effect of Attention Network in Modeling Personal Preference

We show the effect of the attention network in modeling personal preference, by evaluating the performance of PGT$_{-A}$, a variant of PGT, that gives uniform weights to all hidden states in the session RNN without utilizing the attention network. Columns PGT$_{-A}$ and PGT of Table 3 show the accuracy improvements of PGT by the attention network. MRR@20 of PGT increases by 1.47%, 9.47%, and 1.58% on Adressa 1W, Adressa 10W, and Globo, respectively. HR@5 of PGT increases by 1.97%, 2.25%, and 1.78% on Adressa 1W, Adressa 10W, and Globo, respectively. This shows that the attention network highlights important previous hidden states, leading to a superior accuracy.

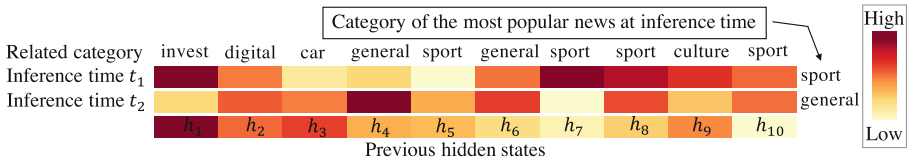


Fig. 3. Attention weights of previous hidden states derived from a sample news watch history. Note that the attention network reacts differently to the same news watch history when the inference time is changed. At times t_1 and t_2 where 'sport' and 'general' categories are popular, respectively, the attention network gives more weights to articles in the same categories.

We additionally perform a case study of the attention network to validate whether it effectively highlights important users' behaviors by considering popular topics. Figure 3 shows attention weights of a sample news watch history at two different inference times t_1 and t_2; a darker cell means a higher weight. At times t_1 and t_2 where 'sport' and 'general' categories are popular, respectively, the attention network gives more weights to articles in the same categories. This result illustrates that PGT dynamically models the personal preference by considering the popular topics at the inference time.

5 Conclusion

We propose PGT, a news recommender system which considers both personal and global temporal preferences to precisely reflect users' behaviors. We observe that the popularity and the freshness of articles, which decay quickly, play important roles in users' watch behaviors. Based on the observation, we introduce the concept of global temporal preference to news recommender system, to provide suitable recommendation results based on time. We also propose an attention based architecture to effectively deal with changes of users' personal preferences, with regard to the global temporal preference. Extensive experiments show that PGT provides the most accurate news recommendation, by considering both of personal and global temporal preferences. Future works include extending the method to handle multiple *heterogeneous* sessions.

Acknowledgments. The Institute of Engineering Research at Seoul National University provided research facilities for this work. The ICT at Seoul National University provides research facilities for this study.

References

1. Gulla, J.A., Zhang, L., Liu, P., Özgöbek, Ö., Su, X.: The Adressa dataset for news recommendation. In: WI (2017)
2. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: ICLR (2016)

3. Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems (2016)
4. Jeon, H., Koo, B., Kang, U.: Data context adaptation for accurate recommendation with additional information. In: IEEE BigData (2019)
5. Khattar, D., Kumar, V., Varma, V., Gupta, M.: HRAM: a hybrid recurrent attention machine for news recommendation. In: CIKM (2018)
6. Khattar, D., Kumar, V., Varma, V., Gupta, M.: Weave&rec: a word embedding based 3-D convolutional network for news recommendation. In: CIKM (2018)
7. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP (2014)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
9. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. Morgan Kaufmann Publishers Inc. (1995)
10. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML (2014)
11. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: SIGKDD (2017)
12. Park, K., Lee, J., Choi, J.: Deep neural networks for news recommendations. In: CIKM (2017)
13. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: LREC (2010)
14. de Souza Pereira Moreira, G., Ferreira, F., da Cunha, A.M.: News session-based recommendations using deep neural networks. In: DLRS@RecSys (2018)
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
16. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: WWW (2018)
17. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: SIGKDD (2019)