Deep Neural Networks for News Recommendations

Keunchan Park
NAVER Corp.
Seongnam, Korea
keunchan.park@navercorp.com

Jisoo Lee
NAVER Corp.
Seongnam, Korea
jisoo.h.lee@navercorp.com

Jaeho Choi
NAVER Corp.
Seongnam, Korea
choi.jaeho@navercorp.com

ABSTRACT

A fundamental role of news websites is to recommend articles that are interesting to read. The key challenge of news recommendation is to recommend newly published articles. Unlike other domains, outdated items are considered to be irrelevant in the news recommendation task. Another challenge is that the recommendation candidates are not seen in the training phase. In this paper, we introduce deep neural network models to overcome these challenges. we propose a modified session-based Recurrent Neural Network (RNN) model tailored to news recommendation as well as a history-based RNN model that spans the whole user's past histories. Finally, we propose a Convolutional Neural Network (CNN) model to capture user preferences and to personalize recommendation results. Experimental results on real-world news dataset shows that our model outperforms competitive baselines.

KEYWORDS

Recommender Systems; Deep Neural Networks

ACM Reference Format:
Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep Neural Networks for News Recommendations. In *Proceedings of CIKM'17* . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3132847.3133154

1 INTRODUCTION

Online news reading has become common activities as the web provides access to millions of news sources all around the world. In these days, people have access to online news 24 hours a day regardless of where they are because of widespread supplies of mobile devices. However, it is difficult to find interesting news articles since the volume of news production from multiple news sources is huge. Thus, recommending good news articles is crucial for online news providers.

News recommendation have distinctive challenges [8] due to the intrinsic characteristics of news domain compared to other domains (e.g., movie, music, products). First, the number of items (i.e., news articles) are very high. New events keep happening and news articles are published every second. Moreover, same contents are published by various news sources and they are treated as different items by the recommender system. Therefore, recommendation approaches that assume fixed item set are unsuitable (e.g., neural networks with softmax layer). Likewise, widely used one-hot representation for news article is impractical. Second, recency is much more important in the news domain than in other domains as its name implies "news" connotes the representation of new information. In most cases, people expect fresh news instead of outdated ones. Consequently, the recommendation candidates should be newly published articles, which means that the candidates are not in the training set. In many cases, new articles face cold-start problem since they do not have any clicks yet. In this situation, co-occurrence based methods such as item-based collaborative filtering cannot recommend new articles. Lastly, news domain covers a wide spectrum of topics and people rarely read articles that are not related to their personal interests and preferences. Hence, it is hard to achieve user satisfaction without personalization.

In this paper, we introduce deep neural network models to overcome aforementioned challenges. To reduce the input dimensionality, we use document embeddings [6] instead of one-hot encoding for news article representation. To capture dynamically changing user interests, we propose two different Recurrent Neural Networks (RNN). One, the session-based RNN model devised by [3] but with a modified ranking loss suited to our setting. The other, a history-based RNN model that considers not only the current session but also past histories of the user. For personalization, recommendation results are reranked by the user's long-term categorical preference. We propose a novel Convolutional Neural Network (CNN) based approach to accumulate long-term user preference. Our experiments on real-world news search click dataset from a commercial search engine reveal that, our proposed deep neural network models are effective than competitive baselines. Moreover, to the best of our knowledge, this is the first attempt to apply deep neural networks for news recommendation.

The remainder of the paper is structured as follows. Section 2 describes our deep neural network models in detail. Section 3 presents experimental results on real-world news dataset. Section 4 briefly summarizes related work. Finally, section 5 concludes with some future research directions.

2 MODEL

2.1 RNN for Recommendation

Suppose there is a sequence of news documents clicked. Let this sequence of length k be $[D(1), D(2), .., D(k)]$. Each $D(t)$ (where $0 \leq t \leq k$) is a bag-of-words representation of t-th clicked document which consists of the words from queries, titles and contents of the news articles. We normalized and lemmatized the word for the semantic equivalence by using in-house tokenizer. In order to get distributed representation of the documents, we trained PV-DBOW

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133154

model [6] from NAVER[1] search engine's data collection. Using this model, we can infer the document vector, $I(t) \in \mathbb{R}^n$ from $D(t)$. We adopted RNN model since we need to capture the dynamics of user interests in search session. The sequence of documents clicked before the last click, $[I(1), I(2), \ldots, I(k-1)]$ is used as inputs to the RNN model. Long Short-Term Memory (LSTM) cell has been used for the hidden cell of RNN structure to deal with vanishing gradient problem. The state of the hidden units of RNN at time step t, $h(t) \in \mathbb{R}^m$ is computed as

$$h(t) = f(I(t) + h(t-1)R^T)$$

where $f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$ is tanh function, used as an activation function. The recurrent connection R encodes the sequential dependencies between documents of consecutive clicks. The state of the final time step of the input, $t = k-1$, $h(k-1)$ can be viewed as features related with user context at time step $k-1$. We need to project this feature vector on document embedding space to get our predicted embedding, $O(k-1)$, and $O(t)$ is defined as

$$O(t) = h(t)U^T + b$$

where $U \in \mathbb{R}^m \times \mathbb{R}^n, b \in \mathbb{R}^n$. We expect that this predicted document vector, O(k-1), is close to the true embedding, which is the embedding of the document last clicked in the session, $I(k)$. The error of model estimation can be defined in a number of ways. We used both cosine and euclidean distance as measures that quantify the errors to be reduced. The objective function is to minimize the mean squared errors (MSE) over training data. We also adopted pairwise ranking losses as suggested in [3]. We modified BPR, TOP1 loss function tailored to our model. Let $S(x, y)$ denote the similarity between x and y. BPR loss is defined as

$$L_{bpr} = -\frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \log(\sigma(S(I_k, O_{k-1}) - S(I_j, O_{k-1})))$$

where N_S is the sample size and I_j is the embedding of the document sampled out of ones that do not exist in same sequence. Top1 loss is defined as

$$L_{top1} = \frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \sigma(S(I_j, O_{k-1}) - S(I_k, O_{k-1})) + \sigma(S(I_j, O_{k-1})^2)$$

which is the same objective function as BPR with addition of a regularization term to lower similarity between prediction and negative samples. We simply used cosine similarity loss functions above. The network is trained with Stochastic Gradient Descent (SGD) via Backpropagation Through-Time (BPTT). In order to minimize failures during training steps, we applied batch normalization [4] and dropouts [11] with probability 0.5 to the network.

We trained two RNN models, session-based and history-based RNN models. We divided user's history into sessions based on 30 minutes of inactivity threshold.[2] In the session-based model, clicks in a session of a user are used as an input whereas all clicks of a user are used as an input to the history-based model.

[1]http://www.naver.com
[2]http://en.wikipedia.org/wiki/Session_(web_analytics)

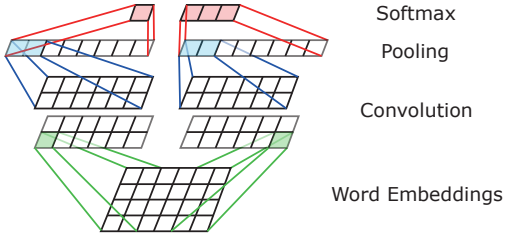

Figure 1: 2-depth CNN classifier

Table 1: News category classification accuracy

Depth	Categories	Train	Test	Accuracy
1	8	3,000,000	150,000	0.90
2	62	586,500	58,650	0.82

2.2 Personalization

Since the spectrum of news topics are wide and user interests differ, personalization is mandatory. In this section, we describe our personalization approach using news categories. People have categorical preferences so that one who reads many articles about sports may never click ones about politics. However, his preference may change over time such that he may start to read political news. In this section, we will introduce methods to build user's categorical preferences to be used for personalized recommendation.

News categories are often categorized into hierarchical structures. For example, NAVER News[3] has two-depth categories such as "Sports>NBA", "Politics>Parliament". The first depth has 8 categories and the second depth has 62 distinct categories. We represented the category of each news article as a concatenation of two vectors of one-hot encoding. One is of length 8 related to first depth category and the other is of length 62 related to second depth. We generated the user preference by wegihted sum of this categories of news that the user has seen. The weight of the article is exponentially decayed as the number of documents in successive clicks increases such that it penalizes the old articles while boosting the recent ones.

The limit of this approach is that not all news articles have assigned categories. Only one-third of news articles have two-depth categories. To exploit data not having category information, we train a category classifier based on CNN. We use the CNN architecture introduced in [5], which has been proven to perform well on text classification problems. The network is composed of four layers: word embedding layer, convolutional layer, pooling layer and fully connected layer with dropouts. The architecture is depicted in Figure 1. Since we have two-depth categories, we train separate CNNs for each depth. In order to prevent the noise from classifier, we update the user's categorical preference only if both model agree with the prediction of depth-1 category. For instance, if depth-1 model predicts "Sports" and depth-2 model predicts "Entertainment>Movie" for a news click, this click does not affect the user's categorical preference. Table 1 shows the accuracy of the

[3]http://news.naver.com

Table 2: Statistics of news session data

Data	Duration	Sessions	Users	Articles
Train	4/22-4/28	1,400,000	8,425,346	5,416,713
Valid	4/22-4/28	100,000	100,000	322,601
Test	4/29,4/30	100,000	95,508	20,063

Table 3: Performance of different models in terms of recall@20, MRR@20 and precision@1

Model	Loss	R@20	MRR@20	PD1@1	PD2@1
POP		0.0474	0.0138	0.7140	0.2874
KNN		0.4509	0.1280	0.8401	0.4212
Session	**BPR**	**0.4875**	**0.1393**	**0.8468**	**0.4201**
Session	TOP1	0.4859	0.1372	0.8467	0.4154
Session	Cosine	0.4588	0.1283	0.8397	0.4231
Session	MSE	0.4562	0.1271	0.8327	0.4242
History	BPR	0.4307	0.1260	0.8364	0.4027
History	TOP1	0.4298	0.1251	0.8360	0.4001
History	Cosine	0.4195	0.1188	0.8251	0.3986
History	MSE	0.4186	0.1145	0.8244	0.3984

classifier for each depth with data statistics. We trained and tested with random sampled news articles that have both categories.

The whole recommendation process is as follows: (1) Session-based RNN retrieves newly published articles that are relevant to the user's immediate interests by feeding in user's current session to the network. (2) History-based RNN retrieves newly published articles that are relevant to the user's short-term interests by feeding in user's past histories to the network (if available). (3) Finally, we rerank the retrieved articles from RNNs by incorporating the similarity between the long-term user preference and the categories of the candidates. If news categories are not available, we use the predicted category by the CNN classifier.

3 EXPERIMENTS

3.1 Dataset

In this section, we evaluate the proposed approaches with NAVER News dataset[4]. Raw data consists of {encrypted login ID, session ID, timestamp, search query, news article ID} for each news click. We collected one month (April 2017) of news search sessions from NAVER. We reserved sessions from the last two days (i.e., 4/29 and 4/30) as test set. Average length of a session in the training set is 2.5. We also filtered out sessions that have clicks of articles published before 4/29, to make sure that evaluation is done on unseen new articles. In total, the number of articles in this test set are 20,063, from which we select for the recommendation.

3.2 Evaluation

The evaluation is done by providing the article of a session in the test set one-by-one and comparing the next articles with the predictions. Specifically, the document embeddings are fed in to the RNN and top-K nearest articles were retrieved from the recommendation candidates based on the output of RNN. We use recall@20, MRR@20 as our primary evaluation metric similar to [3]. As mentioned before, same news are published by many different publishers. Recall and MRR metrics judges wrong even if the predicted article and correct article have identical content. To partially address this problem, we measure precison@1 for news categories on each depth. Although precision on categories is not a precise measure for topical equivalence, we included for a reference.

[4]The dataset cannot be publicized due to proprietary reason.

Table 4: Fine-tuned hyperparameters

Embedding	Dimension	200
Embedding	Context window	5
Embedding	Model	dbow
CNN	Filter Size	2, 3, 4, 5
CNN	Num Filters	100
CNN	Sequence Length	300
RNN	Hidden Size	1000
RNN	Num Steps	10

3.3 Experimental Results

We trained RNN models with sessions of the last week right before the period of test set starts. User preference vectors were accumulated with all sessions except the last two dates (4 weeks). We also split a small portion from the training set as validation set. Detailed statistics are listed in Table 2. We implemented and tested our models with Tensorflow-1.0 [1]. Document embeddings and word embeddings were trained with Gensim [9]. All neural network models were trained on a single Tesla M40 GPU.

3.3.1 Recommendation. Table 3 shows the effectiveness of individual recommendation models. **POP** is the popularity model that recommends 20 most popular articles in the recommendation candidates. Since important headline news is more likely to be read than other articles, it is a strong baseline. **KNN** (K-Nearest Neighbors) model recommends similar articles of the previous article. This is indeed a strong baseline because people read similar articles within a session utill their information need is satisfied. We used cosine distance to calculate the similarity between the candidate embeddings and the previous news article. We compare session-based RNN and history-based RNN separately along with various loss functions in Table 3. We fine-tuned hyperparameters of the RNN models based on the hyperparameters of [3] and listed in Table 4. Several noteworthy points to be mentioned are as follows. As highlited in Table 3, session-based RNN with the modified BPR loss performed best in all measures. BPR loss performed better than TOP1 in most cases. Cosine distance and mean squared error performed worse than BPR and TOP1, but took less time to train due to the absence of negative sampling. LSTM cell performed better than GRU cell when layer normalization was applied. History-based RNN performed worse even than the KNN baseline. Increasing the number of steps to unroll from 10 to 50 neither helped. This is slightly contrary to [10] where history contirbuted to better news ranking. We think that the period of our training data wasn't long enough (i.e., 1 week) to represent long term interests. Still, we found positive examples

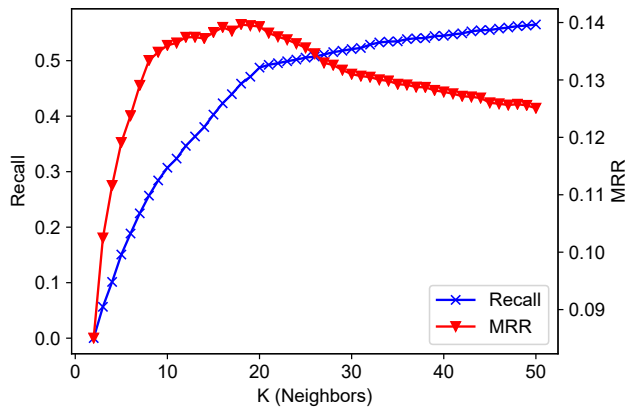

Figure 2: Tradeoff between recall and MRR when reranking

where history-based RNN is useful. If an user has a consistent interest for a long period, history-based RNN recommends relevant articles to the long-term interest even if he/she read articles of completely different topics recently.

3.3.2 Personalization. Figure 2 shows the recall and MRR curve when reranking based on user's categorical preference has been been applied. We increased the number of nearest neighbors retrieved from RNNs to find the optimal K. We tested with the best performing session-based RNN model. It is obvious that recall is proportional to the number of K but the slope declines. However, MRR reaches it's peak at K=18 and declines afterwards. So there is clearly tradeoff between recall and precision. As we increase K, the chances that desirable article will be recommended increase although it is more likely to choose irrelevant articles as recommendation candidates. Thus, it is important to find the optimal K to achieve user satisfaction.

4 RELATED WORK

Liu et al. [7] proposed a news recommender system which predicts user interest via Bayesian Framework, and recommends news articles through collaborative filtering method. However, CF is incapable of recommending news articles that has not been clicked yet. Recently, RNN model has been applied to session-based recommendation tasks. Zhang et al. [13] used RNN model to predict sequential clicks on e-commerce system in order to provide better ad impression. Hidasi et al. [3] applied GRU-based RNN model with pairwise ranking loss function to session-based recommendations of movies and videos. They showed a significant improvement over traditional collaborative filtering approach. Tan et al. [12] demonstrated techniques to further improve RNN application of session-based recommendations. They applied data augmentation and pre-training to account for temporal shifts in the data distribution. They also studied the effects of generalized distillation and directly learning embeddings.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed deep neural networks to the task of presonalized news recommendation. Our models are suitable to real-world news domains, where the number of articles are huge and newly published articles are the recommendation candidates. Two types of RNN models were proposed to model the dynamics of user interests: session-based and history-based model. These models were trained to minimize modified sampling-based ranking losses to suit our setting. Finally, we introduced a personalization approach using categorical preferences. A series of experiments showed effectiveness of our method over competitive baselines.

In the future, we are planning to extend our work in a number of ways: (1) In this work, we assumed all news clicks are positive, but not all clicks are necessarily positive. We are planning to incorporate the dwell time in our RNN models with attention mechanism [2]. (2) Some news articles are not appropriate to recommend. Especially it can give a bad user experience if the quality of the article is bad. We are considering to train a CNN model that assures the article quality. (3) We only used categorical preferences of documents for user profiles in this work. We plan to use other information like demographics and time.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. Tensorflow: large-scale machine learning on heterogeneous systems. (2015). http://tensorflow.org

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[4] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[5] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[6] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*.

[7] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*.

[8] Özlem Özgöbek, Jon Atle Gulla, and Riza Cenk Erdur. 2014. A survey on challenges and methods in news recommendation. In *WEBIST*.

[9] Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://radimrehurek.com/gensim

[10] Vinay Setty, Abhijit Anand, Arunav Mishra, and Avishek Anand. 2017. Modeling event importance for ranking daily news events. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM)*.

[11] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014).

[12] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*.

[13] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*.