Learning Deep Matching-Aware Network for Text Recommendation using Clickthrough Data

Haonan Liu, Nankai Lin, Zitao Chen, Ke Li and Shengyi Jiang✉
School of Information Science and Technology
Guangdong University of Foreign Studies
Guangzhou, China
haonan.liu.edu@gmail.com, jiangshengyi@163.com

***Abstract*—With the trend of information globalization, the volume of text information is exploding, which results in the information overload problem. Text recommendation system has shown to be a valuable tool to help users in such situations of information overload. In general, most researchers define text recommendation as a static problem, ignoring sequential information. In this paper, we propose a text recommendation framework with matching-aware interest extractor and dynamic interest extractor. We apply the Attention-based Long Short-Term Memory Network (LSTM) to model a user's dynamic interest. Besides, we model a user's static interest with the idea of semantic matching. We integrate dynamic interest and static interest of users' and decide whether to recommend a text. We also propose a reasonable method to construct a text recommendation dataset with clickthrough data from CCIR 2018 shared task Personal Recommendation. We test our model and other baseline models on the dataset. The experiment shows our model outperforms all the baseline models and a state-of-the-art model, and the F1-score of our model reaches 0.76.**

Keywords-text recommendation; semantic matching; deep learning

I. INTRODUCTION

The Internet has brought convenience for the dissemination of text information. Text information has been widely spread through online platforms such as Weibo, news portals and knowledge communities. People's access to information has gradually changed from traditional paper media reading to electronic media reading. Obtaining text information becomes easier than before. However, excessive information can also cause user information overload issues. In this case, leveraging text recommendation system can alleviate the user's burden of filtering information. Researchers from industry and academia have long widely studied text recommendation task and proposed many methods. In this paper, we propose a reasonable text recommendation dataset construction method using clickthrough data as well as a content-based text recommendation framework.

Unlike traditional recommendation systems, text recommendation is primarily subject to three challenges. First, text information is time-sensitive. While the volume of texts increases, the timeliness of text decreases over time, which makes methods like collaborative filtering have poor performance on this task. Second, the user's interest will change over time. The text recommendation system should involve the sequential information of users' reading behavior. Third, the interest of users in reading texts is diversified. Users will read text information in multiple fields, and the text recommendation system needs to take the interests of users in various aspects into account to make the best recommendations.

Current text recommendation methods divide into three categories: collaborative filtering methods, content-based methods, and hybrid methods. Collaborative filtering methods [1] depends on the texts read by all users, and learn the features of users relying on the "wisdom of the crowd" to make a recommendation, without using the specific content of the text. Such methods usually suffer from the cold-starting problem. Researchers also propose many content-based methods [2][3][4] due to the rich information in the text content. These methods make a recommendation based on the similarity between texts, but most of them ignore the sequential information of user behaviors. The hybrid method [5] takes the user's information and text information into account. However, most online text information platforms can be accessed anonymously nowadays. In this case, the hybrid method does not perform the text recommendation task well. Most of the researches regard text recommendation as a static problem, ignoring the sequential features of user behavior. Recently, Zhu [6] proposed a content-based news recommendation method Deep Attention Neural Network (DAN). In their work, they involve the user sequential features and achieved state-of-the-art. However, their work does not well preserve the user's static interest features. In the experiments that follow, our model will make a comparison to DAN.

Encountered the challenges in the field of text recommendation and inspired by previous research, we propose a method for constructing a text recommendation dataset that simulates real-world scenarios as well as propose a framework that involves user dynamic interest and user static interest, namely Deep Matching-Aware Network (DMN). We build a dataset by using the clickthrough data of Chinese famous knowledge platform Zhihu from CCIR2018 shared task Personal Recommendation. In our work, DMN uses a convolutional neural network to learn the semantic feature. We implement an Attention-based Long Short-Term Memory Network to learn user dynamic interest features, and the user static interest features are extracted through using methods inspired by semantic matching. With dynamic and static interest features, we predict the probability that a candidate text is recommended.

We use different models to conduct experiments on the dataset. The experimental results show that the proposed DMN model has better results than other models. The remaining part of this paper organizes as follows: Section 2 introduces related studies; Section 3 provides information on our data and describes our data construction method; Section 4 introduces the framework of DMN; Section 5 introduces training of our framework; Section 6 introduces our experiment and analyzes the results of the different

978-1-7281-5014-7/19/$31.00

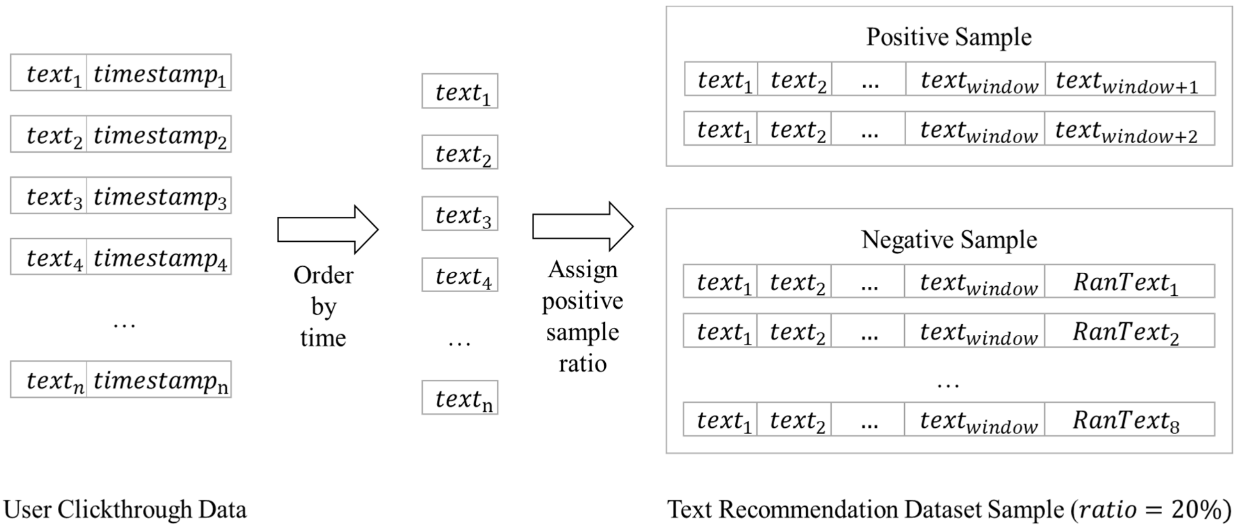

Figure 1. Simple visualization of the data construction method.

models; Section 7 is a case study and explores some variants' influence on our framework; Section 8 concludes our work.

II. RELATED WORK

A. Deep Recommendation System

Deep learning has been successfully applied in many fields, such as computer vision [7], speech recognition [8]. Among these applications, CNN and RNN are the two most popular deep learning methods. Recently, deep neural networks have representative progress in recommendation system. In general, there are two types of deep learning recommendation systems, one for learning the feature of users or items, and the other for simulating the interaction between users and items. For example, the DSSM [9] is used to learn the features of items, and Collaborative Deep Learning [10] learns the interaction features between users and items. Our model DMN learns the interaction features of users and items and also preserve the features of items.

B. Text Recommendation

Text recommendation tasks have long been widely studied, and the tasks are designed to provide users with highly relevant text or to learn user browsing preferences. Examples of text recommendations include recommend blogs [11], social media information [12], news [13], products (based on reviews) [14], and research papers [15]. Due to the rich information in text, many content-based text recommendation methods have been proposed, which are based on the similarity between texts. Zhu proposed a content-based framework DAN [6], using ARNN to capture sequence information and using the attention mechanism to capture the user's static interest feature. Different from DAN, we model the user's static interest based on the semantic similarity method and improve the performance.

C. Semantic Matching

Semantic matching tasks are one of the critical applications of many NLP applications, such as information retrieval, question and answer systems, text recommendations. With the development of deep learning, scholars have proposed several deep learning methods to obtain the semantic information and judge whether the sentence pairs match by comparing the similarity of semantic vectors. DSSM [9], ARC-II [16], DeepMatch [17] are some famous methods. DSSM projects the query and document into the vector space using a nonlinear method. Based on the semantic vector space, the model calculates the similarity between the query and the document. The ARC-II model uses a neural network model to extract information from sentence pairs interactions and to calculate the degree of matching. DeepMatch uses a topic model built with deep architecture to compare similarities between texts. Inspired by the semantic matching task, our work proposes the Matching-Aware Interest Extractor.

III. DATA DESCRIPTION

The data is from CCIR 2018 shared task Personal Recommendation of Zhihu[1]. Zhihu is a famous Chinese knowledge platform, users share knowledge and discuss current affairs on the platform. The full dataset includes question information, answer information, user information, and user clickthrough data. We use the user clickthrough data and reconstruct it in our work.

We propose a method for constructing text recommendation dataset using clickthrough data which aims to simulate the user's behavior in a real-world situation. Figure 1 briefly introduces the process of our method. The method origins in Zhu's work [6] and we make some improvement. Given a user's reading history $\{history_1, \quad history_2, \ldots, history_j, \ldots history_n\}(1 \leq j \leq n)$. Each $history_j$ represents the j_{th} record of a user and

[1] https://biendata.com/competition/CCIR2018/

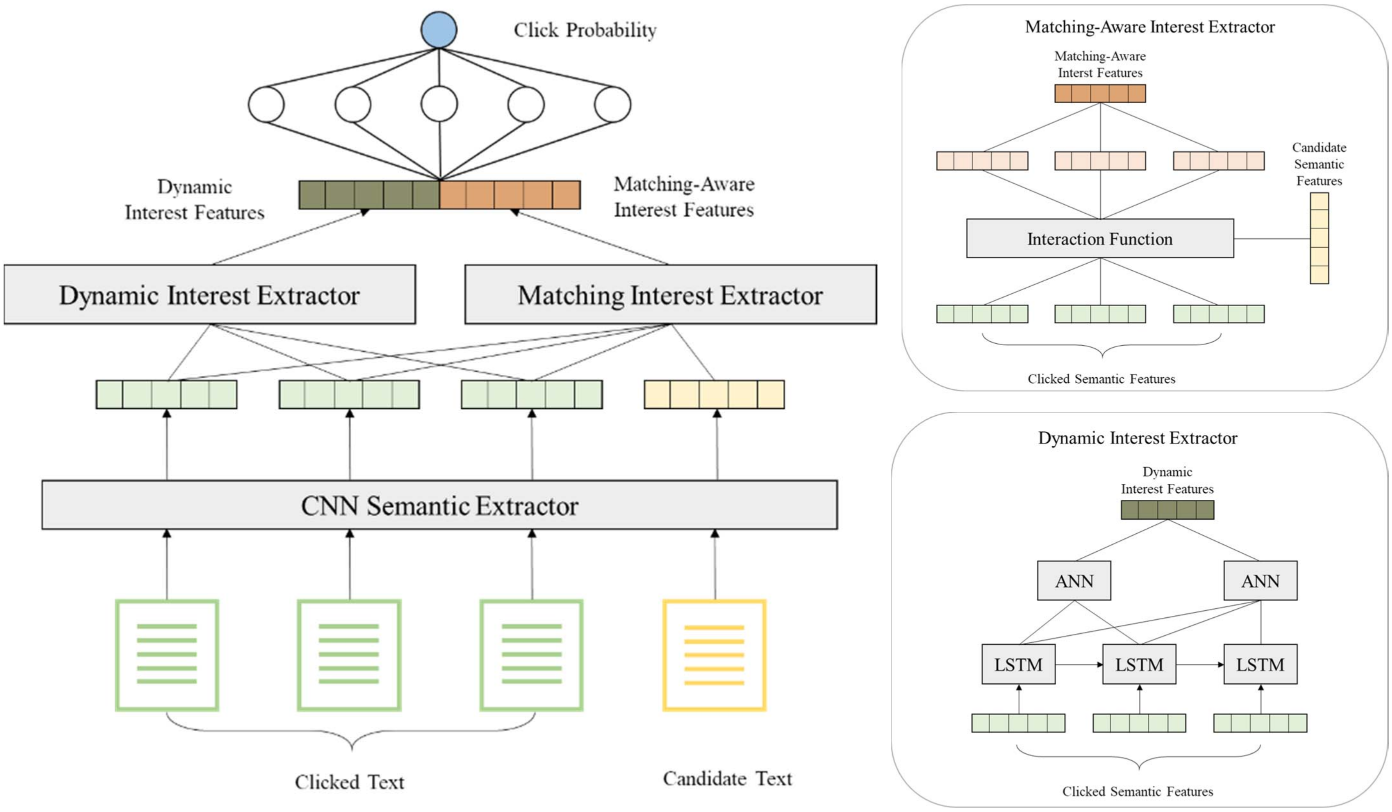

Figure 2. Simple visualization of DMN framework.

comprise of the text content and timestamp. We arrange the user history readings in ascending order of time. The earlier the reading time, the earlier the order and only preserve the text content $text_j$. In other words, the clickthrough data is transform to $\{text_1, text_2, text_3, \ldots, text_n\}$. We set a variant $window$ to intercept the user history into small session $\{text_1, text_2, text_3, \ldots, text_{window}\}$ (window $\in [1, n]$). Each session can represent a user of a range of time. Inspired by the idea of negative sampling in the Word2Vec [18], we propose a random sampling method to construct our data. We assign ten candidate texts for each session, and the positive sample ratio is randomly chosen from the list $[10\%, 20\%, 30\%]$. The positive samples are from the record after window_{th} record and $label_c = 1$. The negative samples are randomly chosen from the text beyond the user's clickthrough data and $label_c = 0$.

We construct the dataset for text recommendation using the beforehand mention method. The training set, development set and test set has 10k,1k,1k samples respectively. Each line has three component: a session that represents a user of a range of time, a candidate text that the user may like, a label that indicates whether the candidate text should be recommended.

IV. DMN Framework

The text recommendation problem described in this paper will satisfy the following definition: Give a user's text reading history, $History = \{text_1, text_2, text_3, \ldots, text_n\}$, $text_j$ is the jth text read by the user, giving a candidate text $text_c$. Each $text_k$ includes word sequence $text_k = \{w_1, w_2, w_3, \ldots, w_n\}$. User's text reading history and the candidate text feed the model, and it will output the probability of recommending the candidate text.

In order to provide a user with suitable texts information, it is necessary to understand a user's long-term interest as well as short-term interest. Our framework adequately models user's interest with a perspective of static and dynamic. Figure 2 shows the structure of the DMN framework. In our model, for input session and candidate text, we use CNN to extract semantic features. Then we learn the user's static interest feature based on the semantic matching method and learn user's dynamic interest by using ARNN. Finally, we predict the probability of recommending the candidate text using a fully connected network.

A. CNN Semantic Extractor

There are many methods to represent text semantic information. Traditional unsupervised text semantic model like Bag Of Word [19] or TF-IDF [20] can not learn information of word order and suffer from data sparsity. Recently, convolution neural network (CNN) have representative progress in computer vision. Researchers proposed many text semantic model [21] based on convolution neural network. In our work, we use CNN to extract semantic information.

CNN is comprised of convolution layer and pooling layer. We denote the embedding input text matrix as $E = \{e_1, e_2, e_i \ldots e_n\}\ (1 \le i \le n)$, e_i indicate i_{th} words embedding vector. We apply filter $k \in R^{d1xd2}$ with a stride $[s_1, s_2]$ to convolution the embedding matrix and receive feature vectors h, which is given by:

$$h = f(E \odot k + b) \quad (1)$$

The symbol ⊙ represents the convolution operation and b is the bias term, $f(\cdot)$ represents the activate function $Relu$. In pooling layer, we apply max pooling operation, we define the max-pooling function as $\mathrm{out} = maxpooling(h, filter_i)$ and we use multiple filters to get multiple output $\mathrm{out_i}$. We concatenate all the output and get semantic vector $S$of text embedding vectors E:

$$\mathrm{S} = [\mathrm{out_1}; \mathrm{out_2}; \dots; \mathrm{out_i}] \quad (2)$$

Symbol $[\cdot;\cdot]$ represents the concatenate of vectors.

B. Matching-Aware Interest Extractor

Matching-Aware Interest Extractor learns user static interest based on semantic matching methods. Each text reading by user interacts with the candidate text and get the interest features.

We get session semantic vectors list $S = [S_1, S_2, \dots, S_n]$ and candidate text semantic vector S_c from CNN semantic extractor. Semantic vector $S_i \in R^n$, n is the dimension of the semantic vector. Each vector from the session will interact with the candidate text semantic vector:

$$f_{inter(i)} = Interaction(S_i, S_c) \quad (3)$$

We experiment with two interaction functions.

Euclidean distance [22], used to calculate the true distance of two points in n-dimensional space. The Interaction function is defined as:

$$Interaction(S_i, S_c) = \sqrt{\sum_k^n (S_{ik} - S_{ck})^2} \quad (4)$$

Cosine similarity [23], is a common function to model interactions. The similarity score is viewed as the angle of two vectors. The Interaction function is defined as:

$$Interaction(S_i, S_c) = \frac{S_i^T v_c}{||S_i|| \cdot ||S_c||} \quad (6)$$

Where $||\cdot||$ stands for the L2 norm.

We concat all the interaction features together as user static interest features:

$$U_{match} = [f_{inter(1)}; f_{inter(2)}; \dots; f_{inter(n)}] \quad (7)$$

C. Dynamic Interest Extractor

In dynamic interest extractor, we use Attention-based LSTM to model user dynamic interest. For each step of the LSTM, we use attention mechanism to dynamically capture user sequential feature. This kind of Attention-based LSTM has applied in many tasks [24][25] and can better capture sequential information. We get session semantic vectors list $S = [S_1, S_2, \dots, S_n]$ from CNN semantic extractor. Semantic vector $S_i \in R^n$, n is the dimension of the semantic vector. We feed the vectors list into LSTM in sequence and get hidden output vector $[h_2, h_3, \dots, h_{n+1}]$:

$$h_i = LSTM(h_{i-1}, S_i) \quad (8)$$

For each hidden vector, we use attention mechanism to calculate user sequential features $f_{atten(i)}$ at this step:

$$f_{atten(i)} = Attention([h_1, \dots, h_{i-1}], hi) \quad (9)$$

With hidden vector list $[h_1, \dots, h_{i-1}]$ and hidden vector h_i at i_{th} step, the attention mechanism is as following:

$$V_j = w_j h_j + b_j \; (j = 1,2, \dots, i-1) \quad (10)$$

$$V_i = w_i h_i + b_i \quad (11)$$

$$\alpha_{j,i} = \frac{exp\left(w_v(V_j + V_i)\right)}{\sum_{j=1}^{i-1} \exp\left(w_v(V_j + V_i)\right)} \quad (12)$$

$$f_{atten(i)} = \sum_{j=1}^{i-1} \alpha_{j,i} h_j \quad (13)$$

Note that V_j represents the j_{th} text hidden vector (j < i), V_i represents the i_{th} text hidden vector and $f_{atten(i)}$ represent user sequential interest features at i_{th} step. We concatenate all the user sequential features together and apply CNN to extract the final user dynamic interest features:

$$U_{seq} = cnn([f_{atten(1)}; f_{atten(2)}; \dots; f_{atten(n)}]) \quad (14)$$

D. Probability Calculator

Finally, we have user sequential interest feature U_{seq} and user static interest feature U_{match}. We concatenate two features, feed it to the fully connected neural network and get the probability $Prob(text_c)$ of recommending the candidate text $text_c$.

V. TRAINING

We denote a training sample as $X = (session, text_c, label_c)$. $session$ is made up of the click history $(\{text_1, text_2, \dots, text_{window}\}$, $text_c$ is the candidate text and $label_c$ is the label. our model will output the probability of recommending $P_c \in [0,1]$. We minimize the following likelihood function to train our model. $c \in \Delta^+$ indicates a positive sample set, and $c \in \Delta^-$ indicates a negative sample set:

$$Loss = -\{\sum_{c \in \Delta^+} ylog(P_c) + \sum_{c \in \Delta^-} (1-y)(1 - log(P_c))\} \quad (15)$$

In order to avoid over-fitting, we apply dropout and L2 regularization to the weight parameters of all the components.

VI. EXPERIMENT

A. Parameter Setting

We implement our model based on Tensorflow[26] and perform multiple experiments with different parameters for each model. We use F1, Precision, and Recall [27][28] as the evaluation metrics. Averaging the results of multiple

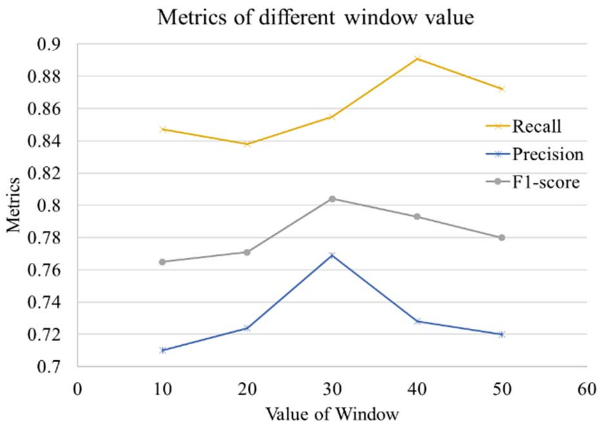

Figure 3. Comparison of different window value.

experiments to compare the performance of different models.

Baseline

We use the following state-of-the-art methods as baselines in our experiments.

- DSSM [9]: is a deep structured semantic model for document ranking. DSSM use a deep neural network to rank a set of documents for a given uery. the user's clicked news is treated as the query and the candidate news are treated as the documents.
- DeepFM [29]: An end-to-end deep learning recommendation model that integrates the architectures of FM and deep neural networks. It models low-order feature interaction like FM and model high-order feature interactions like DNN. We concatenate both clicked texts and candidate text to feed into DeepFM.
- DAN [6]: A deep attention neural network for news recommendation. This model combines user sequential information and user interest feature by Attention and RNN.

Among the three models, DSSM is a model based on semantic matching, DeepFM combines traditional re-commendation method FM and deep learning method DNN. DAN model is a recently proposed model combining sequential and static information.

B. *Result and Analysis*

In the experiment, we compare the DMN model with several benchmark models on our dataset. Table 1 shows the performance of different models.

TABLE I. COMPARISION AMONG DIFFERENT MODELS

Model	Evaluation Metrics		
	Precision	***Recall***	***F1-score***
DSSM	0.692	0.764	0.720
DeepFM	**0.744**	0.695	0.714
DAN	0.748	0.725	0.731
DMN	0.722	**0.818**	**0.761**

TABLE II. COMPARISON OF DIFFERENT INTERACTION METHOD

Model	Evaluation Metrics		
	F1-score	***Precision***	***Recall***
DMN with *Eud*	0.713	0.601	**0.876**
DMN with *cos*	**0.761**	**0.722**	0.818

In the experiment, DeepFM performed poorly compared with other models, F1-score only reached 0.714. F1-score of DSSM model reached 0.720, indicating that the semantic matching method performs well in this task. The DAN model performs the best in the benchmark model with the F1-score of 0.731. Our DMN model performs better than other benchmark models, with the F1-score of 0.761, which is about 0.03 higher than the DAN model. In terms of recall rate, the DMN model achieved the highest recall rate of 0.818, while in terms of accuracy, it was similar to the DeepFM and DAN models. In general, the DMN model can achieve better text recommendation performance compared to other models.

Compared with other benchmark models, DMN has the advantages of (1) using semantic matching to capture user's static interest. (2) Using Attention-based LSTM to learn the user's sequential feature, thereby obtaining user dynamic interest feature (3) We use CNN to extract the semantic features of the text, preserving the word order and semantic information. That is why our model can achieve a better result.

VII. CASE STUDY

A. *Window Variants*

We set the window variable to intercept the user text reading history to represent the user at a certain period. However, we need to consider the effect of the window variable. A small window value may not be able to represent a user. A long window value may not represent a user of a certain period. Thus we choose different values from the list {10,20,30,40,50,60} and conduct experiment. We explore how the window variable affects the recommendation performance. Figure 3 shows that all the metrics increase as the window value grows before window value reaches 30. The highest precision rate and F1-score appear when window value equals 30, while recall rate reaches the highest point with the window value of 40. In conclusion, the model performs the best when the window value equals 30.

B. *Different Matching Method*

Interaction function can learn the similarity of the user's reading text and candidate text. We experimented with two similarity calculation method and compared the performance. Table 2 shows the comparison of the different interaction method. According to the experimental results, using cosine similarity is higher in F1 value and accuracy than using Euclidean distance. In general, using cosine similarity as an interaction method achieve better recommendations performance.

VIII. Conclusion

In this paper, we propose the DMN framework for text recommendation tasks. Unlike other text recommendation models, DMN learns user dynamic interest features and user static interest features from the perspective of semantic matching. We also propose a reasonable data construction method for text recommendation dataset, which simulates the real-world situation, by using clickthrough data. In our experiment, the performance of the DMN model is better than other models, the F1-score reaches 0.76. We also explored how the window length variable affects the result and also experiment our model with different interaction functions in our framework.

In the future, our work will divide into two parts. On the one hand, we will further explore the possibility of applying semantic matching in text recommendation, and try to use a variety of semantic matching methods to learn the user's interest feature. On the other hand, we will consider introducing external knowledge and combine user or knowledge graph information to achieve better performance.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61572145) and the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (No. 2017KZDXM031). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer (Long. Beach. Calif)., 2009.

[2] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization,” in Proceedings of the fifth ACM conference on Digital libraries - DL ’00, 2000.

[3] M. Balabanović and Y. Shoham, “Fab: content-based, collaborative recommendation,” Commun. ACM, 1997.

[4] D. Jannach et al., “Content-based recommendation,” in Recommender Systems, 2013.

[5] V. Vekariya and G. R. Kulkarni, “Hybrid recommender systems: Content-boosted collaborative filtering for improved recommendations,” in Proceedings - International Conference on Communication Systems and Network Technologies, CSNT 2012. 2012.

[6] Q. Zhu, X. Zhou, Z. Song , J. Tan, and L. Guo. “Deep Attention Neural Network for News Recommendation.” Aaai, 2019.

[7] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “Imagenet,” Adv. Neural Inf. Process. Syst. 25, 2012.

[8] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2013.

[9] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” 2013.

[10] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative Deep Learning for Recommender Systems,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’15, 2015.

[11] T. Bansal, M. Das, and C. Bhattacharyya, “Content Driven User Profiling for Comment-Worthy Recommendations of News and Blog Articles,” 2015.

[12] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, “Social media recommendation based on people and tags,” 2010.

[13] O. Phelan, K. McCarthy, and B. Smyth, “Using twitter to recommend real-time topical news,” 2009.

[14] J. McAuley and J. Leskovec, “Hidden factors and hidden topics: understanding rating dimensions with review text,” Proc. 7th ACM Conf. Recomm. Syst. - RecSys ’13, 2013.

[15] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11, 2011.

[16] B. Hu, Z. Lu, H. Li, and Q. Chen, “Convolutional Neural Network Architectures for Matching Natural Language Sentences,” in International Conference on Neural Information Processing Systems, 2014.

[17] Z. Lu and H. Li, “A Deep Architecture for Matching Short Texts,” Adv. Neural Inf. Process. Syst., 2013.

[18] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, “Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[19] Z. S. Harris, “Distributional Structure,” WORD, 1954.

[20] P. K. Gopalan, L. Charlin, and D. Blei, “Content-based recommendations with Poisson factorization,” NIPS Adv. Neural Inf. Process. Syst., 2014.

[21] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[22] J. Asquith and B. Kolman, “Elementary Linear Algebra,” Math. Gaz., 2007.

[23] A. Singhal, “Modern Information Retrieval: A Brief Overview,” Bull. IEEE Comput. Soc. Tech. Comm. Data Eng., 2001.

[24] Y. Wang, M. Huang, xiaoyan zhu, and L. Zhao, “Attention-based LSTM for Aspect-level Sentiment Classification,” 2016.

[25] X. Zhou, X. Wan, and J. Xiao, “Attention-based LSTM Network for Cross-Lingual Sentiment Classification,” 2016.

[26] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” Methods Enzymol., 1983.

[27] Y. Sasaki, “The truth of the F-measure,” Teach Tutor mater, 2007.

[28] D. M. W. Powers and Ailab, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,” J. Mach. Learn. Technol. ISSN, 2011.

[29] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “DeepFM: A factorization-machine based neural network for CTR prediction,” in IJCAI International Joint Conference on Artificial Intelligence, 2017.