Wuhan University
Journal of Natural Sciences

2023, Vol.28 No.1, 029-034

Article ID 1007-1202(2023)01-0029-06
DOI https://doi.org/10.1051/wujns/2023281029

News Recommendation System Based on Topic Embedding and Knowledge Embedding

ZHANG Haojie[1], SUN Hui[2†], QI Baiwen[2], SHEN Zhidong[1,3]

1. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education/School of Cyber Science and Engineering, Wuhan University, Wuhan 430079, Hubei, China;

2. Zhongnan Hospital, Wuhan University, Wuhan 430072, Hubei, China;

3. Engineering Research Center of Cyberspace, Yunnan University, Kunming 650504, Yunnan, China

Abstract: News recommendation system is designed to deal with massive news and provide personalized recommendations for users. Accurately capturing user preferences and modeling news and users is the key to news recommendation. In this paper, we propose a new framework, news recommendation system based on topic embedding and knowledge embedding (NRTK). NRTK handle news titles that users have clicked on from two perspectives to obtain news and user representation embedding :1) extracting explicit and latent topic features from news and mining users' preferences for them in historical behaviors; 2) extracting entities and propagating users' potential preferences in the knowledge graph. Experiments in a real-world dataset validate the effectiveness and efficiency of our approach.

Key words: news recommendation; knowledge embedding; topic embedding; historical behavior

CLC number: TP 399

0 Introduction

Online news websites collect news contents from a variety of sources and provide them to users, attracting a large number of users. However, due to the large amount of news generated every day, it is almost impossible for users to read all the articles. Therefore, it is critical to help users target their reading interests and make personalized recommendations[1-5].

In order to improve the accuracy of recommendation systems, recent research focuses on learning the representation of news more comprehensively. Deep Knowledge-aware Network (DKN)[4] embeds each news from three perspectives: word, entity and entity context, and then designs a CNN model to aggregate these features together. RippleNet[5] obtains the potential interest of users by automatically and iteratively spreading their preferences in the knowledge graph. However, DKN and RippleNet not only ignore rich semantic topics in the news titles, but also fail to consider the relevance be-

Received date: 2022-06-18

Foundation item: Supported by the Key Research & Development Projects in Hubei Province (2022BAA041 and 2021BCA124) and the Open Foundation of Engineering Research Center of Cyberspace(KJAQ202112002)

Biography: ZHANG Haojie, male, Master candidate, research direction: artificial intelligence and big data. E-mail: haojie @ whu.edu.cn

† To whom correspondence should be addressed. E-mail: zn000851@whu.edu.cn

tween topics and users' preferences for those topics to learn more precise news representations.

As shown in Fig. 1, news titles may contain not only a variety of entities, such as politicians, celebrities, companies, or institutions, but also multiple topics, such as politics, entertainment, sports, etc., all of which often play important roles in the title. Long- and short-term user representations (LSTUR) [1] uses explicitly given topic information to learn the representation of news titles. Although explicit topic labels can accurately represent the information of the news, when a news title contains two or more different topics, simple topic information may not be detailed enough to give a more comprehensive representation of the news topic. Therefore, we need the latent topic information to model the news titles in more details.

Fig. 1 Illustration of news title with a variety of entities and topics

For example, the following news title, "Donald Trump *vs*. Madonna: Everything We Know", appears as a music topic. However, the content of the news is more relevant to politics. Such misinterpretation in news modeling can lead to serious errors in learning users' topic preferences. Therefore, only considering the explicit topic information and ignoring latent topic information of news will reduce the accuracy of news recommendation systems.

To address the limitations of existing methods and inspired by the wide success of leveraging knowledge graphs, we propose a news recommendation approach based on topic and entity preference in historical behavior. The core of our approach is a news encoder and a user encoder. In the news encoder, we jointly train news title and word vectors to get the topic information of the news and extract entities to construct the knowledge graph. In the user encoder, we use a combination of long short-term memory network and self-attention mechanism to mine users' topic preferences and a graph attention algorithm to mine users' potential preferences for the entities in knowledge graph based on users' historical behavior. Extensive experiments on a real-world dataset prove the validity of our news recommendation method.

1 Our Approach

In this section, we first introduce the overall framework of news recommendation system based on topic embedding and knowledge embedding (NRTK), as illustrated in Fig. 2, then discuss the process of each module with encoders. NRTK contains three parts, news encoder, user encoder and click predictor. For each news, we extract a news representation vector through the news encoder, which uses two modules to extract features of the news, allowing us to obtain embedding vectors set for a user' s clicked news. In the user encoder, we use the long- and short-term memory network (LSTM) combined with self-attention to learn the user's topic preferences, and then use a graph attention algorithm to aggregate the user's entity preferences to obtain the final representation of the user. In the click predictor, we use the scoring function to calculate the probability of a user clicking the candidate news.

1.1 News Encoder

The news encoder module is used to learn news rep-

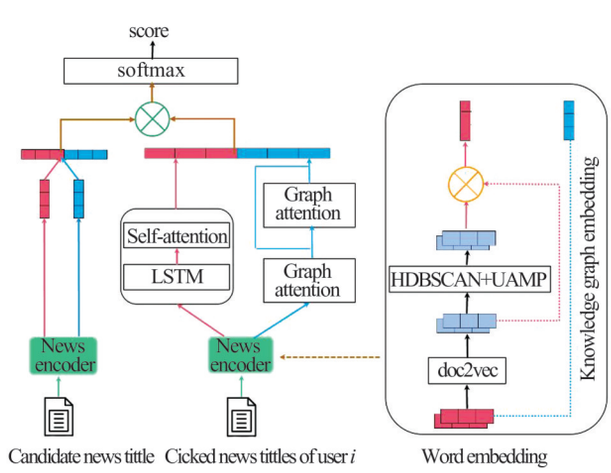

Fig. 2 The framework of our NRTK approach

resentations from news titles. It contains two modules. The first one is word embedding and knowledge graph embedding. Each news title is composed of a sequence of words, $\boldsymbol{t} = [w_1, w_2, \cdots]$. In order to construct the semantic space, we use the word2vec model to pretrain a matrix $\boldsymbol{W}_{n,m}$ for word vectors and a matrix $\boldsymbol{W}'_{n,m}$ for context word vectors. In addition, each word w may be associated with an entity e in the knowledge graph, then we use the TransE[6] to obtain entity embeddings $\boldsymbol{e} \in \boldsymbol{H}^d$, d is the size of the vectors to be learned for each entity in news title, and take the average value $\boldsymbol{k}$ as the knowledge graph embedding of the title.

The second module is topic-level embedding. We use doc2vec Distributed Bag of Words (DBOW)[7] to learn jointly embedded news title and word vectors. The doc2vec DBOW model consists of a matrix $\boldsymbol{D}_{c,m}$, where c is the number of all news titles and m is the size of the vectors to be learned for each news title. For each news title $\boldsymbol{t}$ in the corpus, the context vector $\boldsymbol{w} \in \boldsymbol{W}'_{n,m}$ of each word in the news title is used to predict the news title's vector $\boldsymbol{t}' \in \boldsymbol{D}_{c,m}$. The prediction is softmax $\left(\boldsymbol{w} \cdot \boldsymbol{D}_{c,m}\right)$.

In the learning process, the news title vectors are required to be close to the word vector of the words in them, and far from the word vector of the words not in them. This results in a semantic space where news titles are closest to the words that best describe them and far from words that are dissimilar to them. In this space, an area where news titles are highly concentrated means that news titles in this area are highly similar. This dense area of news titles indicates that these news titles share one or more common latent topics. We assume that the number of dense areas is equal to the number of topics.

We use the uniform manifold approximation and projection for dimension reduction (UMAP)[8] to reduce the dimension of the news title vector. Then, we can use hierarchical density-based spatial clustering of applications with noise (HDBSCAN)[9,10] to identify the dense clusters of news titles and noise news titles in the UMAP-reduced dimension, and uses a noise label or a label of dense clusters to mark each news title in the semantic embedding space.

The topic vectors can be calculated by assigning labels to each dense news title cluster in the semantic embedding space. Our method is to calculate the centroid, i.e. the arithmetic means of all news title vectors in the same dense cluster.

Finally, we get a matrix $\boldsymbol{C}_{x,m}$, where x is the number of topics, m is the dimension of the topic vector. For each news title t, we get its topic embedding as follows:

$$\boldsymbol{W}_t = \text{softmax}(\boldsymbol{t}\boldsymbol{C}^{\mathrm{T}}) \tag{1}$$

$$\boldsymbol{t} = \boldsymbol{W}_t\boldsymbol{C} \tag{2}$$

where $\boldsymbol{W}_t$ is a weight matrix of topics, and $\boldsymbol{t}$ is the news title's topic embedding.

The final representation of a news title is the contact of averaged entity embeddings and topic embedding, formulated as:

$$\boldsymbol{r} = \boldsymbol{k} \oplus \boldsymbol{t} \tag{3}$$

1.2 User Encoder

The user encoder module is used to learn the representations of users from their browsed news. It contains two modules.

The first one is topic preference learning module. The purpose of this module is to learn long-term and short-term user topic preferences. Since users have different degrees of interest in each historical click news title, and the attention mechanism can capture the topic that the user is interested in, long and short-term memory network combined with the self-attention mechanism can be used to mine users' topic preferences according to the users' historical click behavior.

From the news encoder, we have got news' topic embedding $\boldsymbol{T}^{c*m}$. Given the user's click historical matrix $\boldsymbol{Y} \in \boldsymbol{T}^{c*m}$, we can obtain the query $\boldsymbol{Q}$, key $\boldsymbol{K}$ and value $\boldsymbol{a}$ in the self-attention mechanism by the nonlinear transformation of click historical matrix $\boldsymbol{Y}$ as follows:

$$\boldsymbol{Q} = \text{ReLU}(\boldsymbol{Y}\boldsymbol{W}_{\boldsymbol{Q}}) \tag{4}$$

$$\boldsymbol{K} = \text{ReLU}(\boldsymbol{Y}\boldsymbol{W}_{\boldsymbol{K}}) \tag{5}$$

where $\boldsymbol{W}_{\boldsymbol{Q}} \in \boldsymbol{T}^{m*m} = \boldsymbol{W}_{\boldsymbol{K}} \in \boldsymbol{T}^{m*m}$ are weight matrices of the query and key. Then, the weight matrix $\boldsymbol{P}$ can be obtained as follows:

$$\boldsymbol{P} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{m}}\right) \tag{6}$$

where $\boldsymbol{P}$ is a similar matrix of click historical matrix $\boldsymbol{Y} \in \boldsymbol{T}^{c*m}$. Finally, the output of self-attention can be obtained by multiplying the similarity matrix $\boldsymbol{P}$ and the history matrix $\boldsymbol{Y}$.

$$\boldsymbol{a} = \boldsymbol{P}\boldsymbol{Y} \tag{7}$$

where $\boldsymbol{a} \in \boldsymbol{T}^{c*m}$ is the user preferences. We average the self-attention results to learn a single attention value.

$$\boldsymbol{p} = \frac{1}{\boldsymbol{c}}\sum_{i=1}^{c}\boldsymbol{a}_i \tag{8}$$

where $\boldsymbol{p}$ is the user topic preference embedding.

The second module is a knowledge graph-level preference propagation module. In the knowledge graph, the head entity is related to many entities through direct or indirect relationships, but the existence of relation-

ships does not mean that users will have the same degree of interest in these entities. This module uses graph attention networks to learn semantic networks.

To describe users' hierarchically extended preferences based on the knowledge graph, we recursively define the set of n-hop relevant entities for user u as follows:

$$E_u^n = \left\{t|(h,r,t) \in G \text{ and } h \in E_u^{n-1}\right\}, n = 1,2,\cdots,H \quad (9)$$

E_u^0 represents the entities contained in the news titles that the user has clicked on in the past.

We then define the n-hop triple set of user u as follows:

$$S_u^n = \left\{(h,r,t)|(h,r,t) \in G \text{ and } h \in E_u^{n-1}\right\}, n = 1,2,\cdots,H \quad (10)$$

where S_u^n are triples associated with the entities in E_u^n.

Given the average value $\boldsymbol{k} \in \boldsymbol{H}^d$ of entity embeddings in user click news titles and the 1-hop triple set $\mathcal{S}_u^1$ of user u, we use an attention mechanism to learn the entities the user prefers.

$$\boldsymbol{x}_i = \text{softmax}(\boldsymbol{k}^{\mathrm{T}} \boldsymbol{R}_i \boldsymbol{h}_i) = \frac{\exp(\boldsymbol{k}^{\mathrm{T}} \boldsymbol{R}_i \boldsymbol{h}_i)}{\sum_{(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) \in \boldsymbol{S}_u^1} \exp(\boldsymbol{k}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{h})} \quad (11)$$

where $\boldsymbol{r}_i \in \mathbb{R}^{d \times d}$ and $\boldsymbol{h}_i \in \mathbb{R}^{d \times d}$ are the embeddings of relation r_i and head h_i, respectively. The $\boldsymbol{x}_i$ can be regarded as the weight indicating the user's interest in the entity h_i under the relation r_i. Users may have different degrees of interest in the same entity with different relations, so taking the relations into account when calculating the weights can better learn the user's interest in entities.

After obtaining the weights, we multiply the tails in S_u^1 with them, and the vector $\mathbf{hop}_1$ can be obtained by linear addition:

$$\mathbf{hop}_1 = \sum_{(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) \in \boldsymbol{S}_u^1} \boldsymbol{x}_i \boldsymbol{t}_i \quad (12)$$

where $\boldsymbol{t}_i \in \mathbb{R}^{d \times d}$ represents the tails in S_u^1. Through this process, a user's preferences are transferred from his click history to the 1-hop relevant entities E_u^1 along the links in S_u^1.

By replacing $\boldsymbol{k}$ with $\mathbf{hop}_1$ in Eq. (11), the module iterates this procedure over user u's triple set S_u^i for $i = 1,\cdots,N$. Therefore, a user's preference is propagated N times along the triple set from his click history, and N different preference sequences are generated: $\mathbf{hop}_1$, $\mathbf{hop}_2$, $\cdots$, $\mathbf{hop}_N$. To represent the user's final entity preference embeddings, we merge all embeddings.

$$\boldsymbol{f} = \sum_{i=1}^{N} \mathbf{hop}_i \quad (13)$$

The embedding $\boldsymbol{f}$ is the output of this module.

The final user representation is the contact of entity preference embedding and topic preference embedding, formulated as:

$$\boldsymbol{u} = \boldsymbol{p} \oplus \boldsymbol{f} \quad (14)$$

1.3 Click Predictor

The click predictor is used to predict the probability of a user clicking a candidate news. Denote the representation of a candidate news t as $\boldsymbol{r}$, the click probability score $\hat{\boldsymbol{y}}$ is computed as follows:

$$\hat{\boldsymbol{y}} = \boldsymbol{\sigma}(\boldsymbol{u}^{\mathrm{T}} \boldsymbol{r}) \quad (15)$$

where $\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{x})}$ is the sigmoid function.

2 Experiments

2.1 Datasets and Experimental Settings

We use the Bing News server logs from May 16, 2017 to January 11, 2018 as our dataset. Each piece of impression in the dataset contains a timestamp, a news ID, a title, a category label. The basic statistics and distribution of the news dataset are shown in Table 1. In our experiments, we divided the dataset into training set, validation set and test set in a 6:2:2 ratio. The word embeddings are 300-dimensional and initialized by the word2vec model. The entity embeddings are 50-dimensional and initialized by the TransE. And we set the hop number H = 2. These hyperparameters are tuned on validation set. In addition, the experiment was independently repeated for 10 times and the average results in terms of area under curve (AUC) and accuracy (ACC) was taken for performance analysis.

Table 1 Dataset statistics

Dataset	Number of dataset
users	132 747
news	511 726
triples	7 689 563
impressions	1 116 589
avg. words per title	10.34
avg. entities per title	3.9

2.2 Baselines

We use the following models as baselines in our experiments: 1) LSTUR[1], a neural news recommendation method; 2) Factorization Machine Library (LibFM)[11], a feature-based factorization model; 3) Deep Structured Semantic Model (DSSM)[2], a deep structured semantic

model; 4) DeepWide[3], a popular neural recommendation method; 5) DeepFM[12], a deep model for recommendation; 6) DKN[4], a deep knowledge-aware network for news recommendation; 7) RippleNet[5], a memory-network-like approach.

2.3 Results

The results of all methods in click-through-rate (CTR) prediction are presented in Table 2. Experimental results show that our recommendation system performs best compared with other recommendation models. Specifically, NRTK outperforms baselines by 1.9% to 8.0% on AUC and 2.1% to 8.3% on ACC, respectively.

We also evaluate the influence of maximal hop number H on NRTK performance. The results are shown in Table 3 which shows that the best performance is achieved when H is 2 or 3. This is because if H is too small, it is difficult to explore the connection and long-distance dependence between entities, while if H is too large, it will bring much more noise than useful signals.

Table 2 The results of AUC and ACC in CTR prediction

Model	AUC	ACC
LSTUR	0.652	0.607
LibFM	0.644	0.588
DSSM	0.634	0.589
DeepFM	0.617	0.569
DeepWide	0.654	0.596
DKN	0.663	0.604
RippleNet	0.678	0.631
NRTK	**0.697**	**0.652**

Table 3 The results of AUC with respect to different hop numbers

Hop-number	1	2	3	4	5
AUC	0.687	0.695	0.697	0.682	0.671

2.4 Ablation Study

To verify the validity of our approach that attention mechanisms can improve recommendation performance, we designed an ablation study to evaluate our model. In this section, instead of using attention mechanisms to capture user preferences for topics and entities, the ablation model simply aggregates them together. The experimental results are shown in Fig. 3. From these results, we find the self-attention and graph attention are very useful. This is because users have different interests on different topics and entities, and capturing users' preferences is important for recommendations.

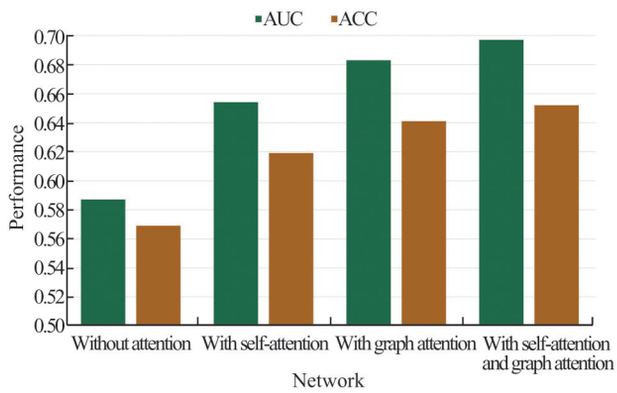

Fig. 3 Effectiveness of different attention networks

2.5 Parameter Sensitivity

In this section, we study the effect of parameters d and training weight of knowledge graph embedding term λ_2 on the model performance. We change d from 2 to 128 and λ_2 from 0 to 1.0, keeping other parameters constant. The results of AUC are shown in Fig. 4. We observe from Fig. 4(a) that the performance of the model improves at the beginning with increasing d, as larger dimensional embeddings can encode more useful information, but degrades after d = 64 due to possible overfitting. From Fig. 4(b), it can be seen that the performance of NRTK reaches the best when $\lambda_2 = 0.01$.

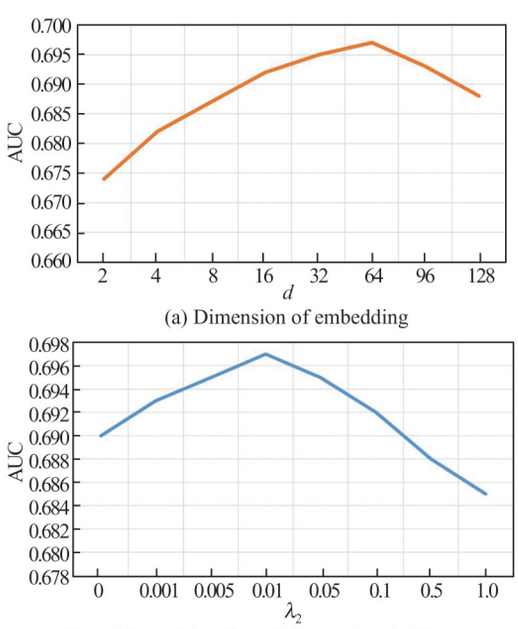

Fig. 4 Parameter sensitivity of NRTK

3 Conclusion

In this paper, we propose NRTK, an end-to-end framework that naturally incorporates the topic model and knowledge graph into recommendation systems. NRTK overcomes the limitations of existing recommendation methods by addressing two major challenges in news recommendation: 1) explicit and latent topic features are extracted from news titles by topic-level embedding, and users ' long-term and short-term preferences are mined for them; 2) through knowledge graph-level preference propagation module, it automatically propagates users ' potential preferences and explores their hierarchical interests in the knowledge graph. We conduct a lot of experiments in a recommendation scenario. The results show that NRTK has a significant advantage over the strong baseline.

For future work, we plan to improve the efficiency and precision of finding topics and further investigate the methods of characterizing entity-relation interactions.

References

[1] An M X, Wu F Z, Wu C H, *et al*. Neural news recommendation with long-and short-term user representations[C]// *Proceedings of the* 57*th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 2019: 336-345.

[2] Huang P S, He X D, Gao J F, *et al*. Learning deep structured semantic models for web search using clickthrough data[C]// *Proceedings of the* 22*nd ACM International Conference on Information & Knowledge Management*. New York: ACM, 2013: 2333-2338.

[3] Cheng H T, Koc L, Harmsen J, *et al*. Wide & deep learning for recommender systems[C]// *Proceedings of the* 1*st Workshop on Deep Learning for Recommender Systems*. New York: ACM, 2016: 7-10.

[4] Wang H W, Zhang F Z, Xie X, *et al*. DKN: Deep knowledge-aware network for news recommendation[C]// *Proceedings of the* 2018 *World Wide Web Conference*. New York: ACM, 2018: 1835- 1844.

[5] Wang H W, Zhang F Z, Wang J L, *et al*. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems[C]// *Proceedings of the* 27*th ACM International Conference on Information and Knowledge Management*. New York: ACM, 2018: 417-426.

[6] Bordes A, Usunier N, Garcia-Duran A, *et al*. Translating embeddings for modeling multi-relational data[C]// *Advances in Neural Information Processing Systems*. New York: ACM, 2013: 2787-2795.

[7] Rehůřek R, Sojka P. Software framework for topic modelling with large Corpora[C]//*Proceedings of the LREC* 2010 *Workshop on New Challenges for NLP Frameworks*. Stroudsburg: Association for Computational Linguistics, 2010: 45-50.

[8] McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction[EB/OL]. [2022-05-18]. http//www. arXiv preprint arXiv: 1802.03426.

[9] Campello R, Moulavi D, Sander J. Density-based clustering based on hierarchical density estimates[C]// *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. New York: ACM, 2013:160 - 172.

[10] McInnes L , Healy L. Accelerated hierarchical density based clustering [C]//2017 *IEEE International Conference on Data Mining Workshops* (*ICDMW*). Washington D C: IEEE, 2017: 33-42.

[11] Rendle S. Factorization machines with libfm[C]// *ACM Transactions on Intelligent Systems and Technology* (TIST). New York: ACM, 2012: 1-22.

[12] Guo H F, Tang R M, Ye Y M, *et al*. DeepFM: A factorization-machine based neural network for CTR prediction[EB/OL]. [2022-05-18]. http//www. arXiv preprint arXiv:1703.04247.

□