KSR: Knowledge-based Sequential News Recommendation System

Jiameng Bai¹², Xin Huang¹², Min Li*¹ and Jia Peng¹

¹State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, 100093, China
²School of Cyber Security, University of Chinese Academy of Sciences, Beijing, 100049, China
*Corresponding author’s e-mail: minli@iie.ac.cn

Abstract. With the popularity of the Internet, some online news reading habits have gradually replaced traditional media devices. People can see a variety of news on mobile phones or web pages. The overloaded information makes it impossible for users to quickly get the news they want to see, so the existence of a news recommendation system is necessary. However, unlike other fields, the recommendation of news is time-sensitive, and the problem of sparse data makes traditional collaborative filtering algorithms invalid. There are many entities in the news, and the knowledge graph is a collection of a large number of entities. In this paper, we propose a KSR model (Knowledge-based Sequential Recommendation) which uses the knowledge graph as side information to enrich the feature representation of the news to calculate the user’s probability to click the forecasted news. KSR uses the knowledge graph to represent the entities in the news and uses a recurrent neural network to capture the sequential relationships in the news data to enrich the feature representation of the news. Because the different news that user has clicked have different relations with the news to be predicted, the attention model is introduced to calculate the weights. Finally, we conduct experiments on the existing dataset and results prove the efficacy of KSR over several baselines. Different contrast experiments also prove the effectiveness of each module of the model.

1. Introduction

With the booming popularity of the Internet and the changes in the way people read, traditional media such as newspapers, radios, and other devices are gradually out of sight. At the same time, some online news reading systems have emerged, such as Google News and Bing News. People can see news from the world on their mobile phones or the web. Massive news gives users a lot of opportunities to choose the information they want. However, the overload of information makes it impossible for users to obtain the information they want quickly. The appearance of the recommendation system dramatically slows down the information overload. It can personalize the recommendation according to the user's historical interest [1] to help user find information quickly.

Different from recommendation systems in other fields, such as movies, music, and e-commerce recommendations. Recommendation systems in the area of news often face the following challenges. First, the timeliness of news recommendation is very high. [2] shows that in business news dataset news will soon become obsolete, and the average life of news is only 4.1h. In the recommendation field, collaborative filtering has achieved great success, and the collaborative filtering method is based on the interaction record of user history, which is heavily dependent on the user's item scoring matrix. It will
face the problem of sparse and cold start. The problem in news domain makes the traditional collaborative filtering recommendation [3] invalid. Second, when users read the news, interest tends to shift significantly with time flows. Third, the news field has a unique feature that the language used in the title is often highly concise and contains many entity words.

In order to solve these problems, researchers are committed to introducing more side information to make recommendations valid, such as users-item attributes [4] and social networks [5]. Among this side information, knowledge graph appears as a relatively new knowledge information in the sight of researchers. Knowledge graph is a graph network that contains rich semantic information. The nodes in the knowledge graph represent the entity, and the edges represent the relationships between the entities. In recent years, researchers have proposed many knowledge graphs, such as DBPedia, Google Knowledge Graph, and Microsoft Satori. These knowledge graphs have been widely applied to the question-and-answer system [6] and the study of the word vector [7].

Inspired by the idea of applying knowledge graphs to other tasks, as shown in Figure 1, we can also apply the knowledge graph to the recommendation system. Applying the knowledge graph to the recommendation system has the following benefits. First, the knowledge graph contains a large number of semantic relationships between items which can find hidden deep relationships between items and can improve the accuracy of recommendations. Second, the knowledge graph is a heterogeneous graph, which contains many different types of relationships, which can enrich the user's interest portraits and provide more diverse recommendation results. Third, the knowledge graph connects the user history and the objective item, which provides interpretability for recommendations. So we introduce the knowledge graph into the recommendation system and propose the KSR model (Knowledge-based Sequential Recommendation). The KSR model aligns the entity in the news with its entities in the knowledge graph, extracts the subgraphs containing the entities in the knowledge graph, and uses the knowledge graph to represent the learning method for encoding. Since the title’s words of the news are sequential, the vector which training from the knowledge graph is introduced into the recurrent neural network for learning the sequential relationships. The user’s feature is represented by his historical clicked news. Since the relationship between the news to be predicted and the news that the user has seen has different weights, we use the attention model to calculate those weights.

In summary, this article contributes as follows:

(1) We introduce the knowledge graph into the recommendation system. That is, the feature of the news data is represented as its own word vector, and the feature representation of the entity appearing in the news in the knowledge graph which can enrich the feature representation of the news. There are different degrees of correlations between the news clicked by the user history and the news to be predicted, so we introduced the attention mechanism to calculate that weight.
(2) The words in the news’s title contain sequential relationships. The existing knowledge graph-based recommendation model does not capture these relationships. our KSR model uses a recurrent neural network to learn the sequential relationship. Compared with some existing models, it achieves better results. Moreover, in the subsequent comparative experiments, it is also proved that the module for the sequential relationship has a specific improvement for the recommendation effect.

2. Preliminaries

2.1. LSTM cell

The LSTM(Long and Short Time Memory)[8] network is a kind of recurrent neural network that can capture the sequential relationship well. It is a chain network with repeated recurrent neural network cells.

LSTM can well capture sequential relationships, and on the other hand, it can solve long-term dependence problems in recurrent neural networks, such as gradient disappearance or gradient explosion. It solves these problems by introducing a gate mechanism. For example, "Forget Gate", "Input Gate", "Update Gate" and "Output Gate". The Forget Gate controls how much information at the previous moment is retained to the current state. The Input Gate determines how much of the current input is retained. The update gate is used to update the current state. Finally, the Output Gate determines the output of the current layer.

2.2. KGE(Knowledge Graph Embedding Solutions)

A knowledge graph contains a large number of entity-relationship triples \((h, r, t)\), where \(h\), \(r\), \(t\) represent the head node, the relationship node, and the tail node, respectively. Knowledge graph representation learning refers to the embedding of entities and relationships in the knowledge graph. The method is used to represent entities and relationships in a low-dimensional dense vector space. In recent years, translation-based graph representation learning methods have received widespread attention due to their outstanding performance. In this paper, we introduce the following methods:

TransE[9] expects \(h + r \approx t\) when \((h, r, t)\) is present. TransH[10] handles a one-to-many relationship by mapping the representation of the entity to a relational space. TransR[11] proposed to establish their own relational space for different relationships. TransD[12] overcomes the shortcomings of TransR in using the same transformation matrix for mapping. It uses different vector space to map the node and relation.

3. Model Define

In this section, we will introduce the KSR model in detail. The framework of the model is shown in Figure 2. The model takes the news that the user has clicked and the news that is to be predicted as input. For each piece of new we use the KGE module to process its information level representation, and get the embedding of each piece of news. We enrich the news embedding vector by capturing sequential relationship through LSTM, which is described in detail in 3.1. The user's embedding is represented by the news that he has clicked. In order to obtain the final embedding of the news, we use the attention mechanism to calculate the weight of user history vector and the news vector to be predicted which will be described in detail in 3.2. Finally, we put the user's embedding and the embedding of the news to be predicted into the neural network to get the probability that the user clicks on the news.
3.1. News embedding refining

The KGE and LSTM modules work primarily to get the embedding of the news, which is used to enrich the user’s characteristics. First, we use the traditional word2vec for the title in the dataset. We use the Gensim library to calculate the word embedding \( w_i \), as shown in Equation 1. Then we get the entities in the title, and extract all the relationship links from the original knowledge graph to construct a subgraph. For the extracted subgraphs, we use the knowledge graph embedding methods TransE, TransH, TransR, and TransD to get the entity embedding. The entity embedding is expressed as \( e_i \), as shown in Equation 2. We found in the experiment that when the entities in the news titles are constructed into small subgraphs, although the structural information of some entities can be obtained, these subgraphs are often sparse. So we use the context entities to enrich the sematic information. The context entity is represented as the average of its neighbor entities, denoted here by \( c_i \), as expressed by Equation 3. We complete the first step of feature refinement. Next, we will use the recurrent neural network to capture the sequential relationship between each part of the vector.

\[
\begin{align*}
    w_i &= \text{Gensim}(\text{word}_i) \\
    e_i &= \text{Trans}(G_i) \\
    c_i &= \frac{1}{|\text{context}(e_i)|} \sum_{e \in \text{context}(e_i)} e
\end{align*}
\]

We can do this by using the recurrent neural network, as shown in Section 2.1. In subsequent section, we also used the GRU unit which is variants of LSTM for comparative experiments. We pass the word embedding in the title mentioned above and the entity embedding generated by Trans method into two different LSTM networks to obtain a embedding containing the sequential relationship, such as formula (4)(5), where \([w_1 w_2 \ldots w_n]\) is the words contained in the news. The embedding of the entity part is the same. Finally, the three parts of the embedding are connected as an input to the subsequent modules, as shown in Equation 6.

\[
\begin{align*}
    [w_1 w_2 \ldots w_n] &= LSTM(w_1 w_2 \ldots w_n) \\
    [e_1 e_2 \ldots e_n] &= LSTM(e_1 e_2 \ldots e_n) \\
    W &= [w_1 w_2 \ldots w_n e_1 e_2 \ldots e_n c_1 c_2 \ldots c_n]
\end{align*}
\]

3.2. Attention module

Through the processing in Section 3.1, we obtain the news embedding that the user history clicked. We use the news that the user history clicked to indicate the user’s preference. We can average these vectors, but because of the variety of user interests, news that have been clicked by user history may have different effects on the predicted news. In order to indicate the user’s preference for different news, we...
use the attention model [13] to calculate the weight of news clicked by users and news to be predicted. The attention model diagram is shown in Figure 3. We use the news to be predicted, and the news clicked by the user history as input to the deep neural network for training, and finally use the softmax function to learn the weight of each historical news.

![Figure 3. The attention model.](image)

### 4. Experiment

In this section, we will introduce the experimental part, including the introduction of the dataset and the results of the experiment and the comparative experiment.

#### 4.1. Dataset description

Our data set comes from Microsoft Bing News (for publicly released news data). After desensitization, each log mainly includes the user ID, news title and click status (1 is clicked, 0 is unclicked), entity ID, entity name. The knowledge graph is a graph of all the entities present in the dataset in the Microsoft Satori knowledge graph and their neighbours within several hops.

#### 4.2. Baselines

DMF [14] is a depth matrix factorization model that uses a multi-layer neural network to process user embedding and item embedding for recommendation. We only use the user’s implicit feedback for input.

FM [15] is a representative model based on use’s history. We use the doc2vec model to encode the news titles for calculation.

DKN [16] takes entity embedding and word embedding as input and encodes it into the KCNN network for encoding.

#### 4.3. Experimental setup

Our experimental environment is Win7 64-bit, Python 3.6 and TensorFlow 1.14. KGE module is FastTransX. It is an effective way to implement TransE and its variants, compiling with g++. Our optimization algorithm is Adam. The loss function is the cross entropy loss function plus the loss of the regularization term. The optimization goal is shown in Equation 7. Where $y$ is the label of the sample, and $\tilde{y}$ is the probability that the model is predicted to be positive.

$$
Loss = \sum -[y \log(\tilde{y}) + (1 - y) \log (1 - \tilde{y})] + \theta \ast Loss_{regularization} \tag{7}
$$
4.4. Experimental results

We analyze the experimental results with AUC and MSE. AUC is an evaluation value to measure the pros and cons of the model, indicating the probability that the positive example of the prediction is in front of the negative example. Performance of the model is better when this value is higher. MSE is the mean square error, which is the expected value of the square of the difference between the predicted value and the real value, which we expect the lower, the better. As shown in Table 1, the KSR model obtains the highest AUC value and the lowest MSE value and it achieves best results than the baselines.

| Model  | AUC     | MSE      |
|--------|---------|----------|
| KSR    | 58.38   | 0.2406   |
| FM     | 44.96   | 0.2428   |
| DKN    | 57.65   | 0.2419   |
| DMF    | 53.90   | 0.2591   |

Table 1. Comparison of different models

We will analyze the model and conduct comparative experiments from the following four dimensions.

1. In Section 3.1, we mentioned that we not only use the vector of the entity in the news but also introduce the context entity. We will experimentally verify whether this works. 2. There is a well-known variant of LSTM—GRU, we will compare which effect is better. 3. We introduced the attention model into the model. We will verify this in our experiment. 4. In Section 2.2, we introduced four KGE methods, and we will figure out that which one has best effect. We performed experiments on these four cases on the KSR model, and the experimental results are shown in Table 2. The first row in the table is the standard KSR model, using LSTM to capture sequential relationships and using TransD model to get the knowledge graph embedding. The second and third rows in the table prove the necessity of introducing the context vector and attention module. The fourth row in the table is the result of sequential capture by using GRU method, which proves that LSTM achieves better results than GRU. The fifth, sixth, and seventh rows in the table are the results of using different Trans methods, which proves the superiority of using TransD.

| Epoch   | Epoch   | Epoch   | Epoch   |
|---------|---------|---------|---------|
| @5      | @10     | @15     | @20     |
| KSR     | 0.571   | 0.577   | 0.572   | 0.569   | 0.570   |
| KSR(-context) | 0.568 | 0.559 | 0.520 | 0.525 | 0.543 |
| KSR(-Attention) | 0.568 | 0.515 | 0.545 | 0.528 | 0.522 |
| KSR(GRU) | 0.567 | 0.550 | 0.512 | 0.560 | 0.544 |
| KSR(TransH) | 0.564 | 0.559 | 0.556 | 0.557 | 0.465 |
| KSR(TransE) | 0.564 | 0.563 | 0.547 | 0.547 | 0.523 |
| KSR(TransR) | 0.571 | 0.569 | 0.571 | 0.560 | 0.524 |

Table 2. Comparison of variants.

5. Conclusions

This paper proposes a KSR model for the problem of data sparsity in the field of news recommendation. KSR uses a knowledge graph as side information to introduce the recommendation process, enriches the feature representation of news vector, and proposes to introduce the sequential module into the model to capture the sequential relationship between words and entities. Since the relationship between the news clicked by the user history and the new to be predicted has different weights, based on this consideration, we introduce the attention mechanism into the model. Finally, the experiments are carried out on the actual news dataset, it achieves best results than the baselines. In the subsequent experimental analysis, the rationality and effectiveness of the module in this model are also deeply analyzed.
Acknowledgments
I want to thank my mentor, my predecessors and my classmates for their support and help so that I can achieve my present achievements. This work is partially supported by National Key Research and Development Program of China (No.2019QY1704).

References
[1] Son, J. W., Kim, A., Park, S. B. (2013). A location-based news article recommendation with explicit localized semantic analysis. In: SIGIR. Dublin. pp. 293-302.
[2] Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., Li, Z. (2018). DRN: A deep reinforcement learning framework for news recommendation. In: WWW. Lyon. pp. 167-176.
[3] Wang, C., Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In: SIGKDD. San Diego. pp. 448-456.
[4] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In: Neural Information Processing Systems. Montréal. pp. 3111-3119.
[5] Covington, P., Adams, J., Sargin, E. (2016). Deep neural networks for youtube recommendations. In: ACM conference on recommender systems. Boston. pp. 191-198.
[6] Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., et al. (2016). Wide deep learning for recommender systems. In: Deep Learning for Recommender Systems. Boston. pp. 7-10.
[7] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP. Seattle. pp. 1631-1642.
[8] Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N. (2019). SR-LSTM: State Refinement for LSTM towards Pedestrian Trajectory Prediction. In: CVPR. Long Beach, CA. pp. 12085-12094.
[9] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In: NeurIPS. Montréal. pp. 2787-2795.
[10] Wang, Z., Zhang, J., Feng, J., Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In: AAAI. Québec City. pp. 1112-1119.
[11] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In: AAAI. Austin. pp. 2181-2187.
[12] Ji, G., He, S., Xu, L., Liu, K., Zhao, J. (2015). Knowledge graph embedding via dynamic graphing matrix. In: ACL. Beijing. pp. 687-696.
[13] Wang, X., Yu, L., Ren, K., Tao, G., Zhang, W., Yu, Y., Wang, J. (2017). Dynamic attention deep model for article recommendation by learning human editors’ demonstration. In: SIGKDD. Halifax. pp. 2051-2059.
[14] Xue, H. J., Dai, X., Zhang, J., Huang, S., Chen, J. (2017, August). Deep Matrix Factorization Models for Recommender Systems. In: IJCAI. Melbourne. pp. 3203-3209.
[15] Rendle, S. (2010, December). Factorization machines. In: ICDM. Sydney. pp. 995-1000.
[16] Wang, H., Zhang, F., Xie, X., Guo, M. (2018, April). DKN: Deep knowledge-aware network for news recommendation. In: WWW. Lyon. pp. 1835-1844.