2021

Let Knowledge Make Recommendations For You

Zhiixue Jiang

Chengying Chi

Yun Yun Zhan

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Let knowledge make recommendations for you

ZHIXUE JIANG[1],CHENGYING CHI[1], AND YUNYUN ZHAN[2]
[1]University of Science and Technology Liaoning, Anshan, 114031, China
[2]College of Science & Health, Technological University Dublin, Dublin, D08 X622, Ireland

Corresponding author: Chengying Chi(chichengying@ustl.edu.cn)

This research was supported by the National Natural Science Foundation of China (NSFC) under the grant (No.61672138)

ABSTRACT The knowledge graph can make more accurate personalized recommendations for the recommendation system, but it is also interpretative and has traces to follow. The purpose of the recommendation system is to recommend a series of unobserved items for users. At present, recommendation systems based on knowledge graphs are mainly implemented in two ways: Embedding-based and path-based. Embedding methods usually directly use information from the knowledge graph to enrich the representation of an item or user. Still, it failed to introduce multi-hop relations, and it is challenging to use semantic network information. A path-based recommendation algorithm utilizes the knowledge graph to gain multi-hop knowledge and compare the similarity between users or items to improve the recommendation effect. This paper (1) Aiming at the problem of how the recommendation algorithm effectively utilizes the semantically related information of knowledge, a self-attention-based knowledge representation learning model is designed to learn the semantic information of the entity-relationship by using the overall triplet of the entity-relationship to achieve high-quality knowledge features, Which brings more and more helpful information to the recommendation. (2) Constructing a content recommendation model with unified, embedded behavior and knowledge features, using historical user preferences combined with knowledge graphs to dynamically learn knowledge features to bring users more accurate and diverse recommendations. (3) Aiming at the problem of knowledge feature representation learning, a self-attention-based knowledge representation learning model is proposed. Focusing on the difference in the importance of triples for determining entity semantics, the self-attention mechanism is used to learn semantics from triples to improve knowledge features. The quality of the representation provides high-quality auxiliary information for the recommendation system. The model's performance is demonstrated through link prediction and triple classification experiments to prove the feasibility of the method proposed in this article.

INDEX TERMS Knowledge graph, recommendation system, knowledge feature representation

I. INTRODUCTION

A. RESEARCH BACKGROUND, RESEARCH SIGNIFICANCE

The recommendation system technology is a means to solve the overload of Internet information, filtering information for users and recommending the content they are interested in. The core value of the recommendation system is to help users easily extract the content of interest, provide reference opinions when users face unfamiliar areas, and provide help to users when user needs are not precise to satisfy users' curiosity. One of the traditional recommendation systems using the interactive behavior information users leave on the product is a collaborative filtering recommendation algorithm to find users with the same user interests and recommend these users with the same interests to browse items. However, when a user newly joins, the user has no historical behavior, that is, a cold start, so it is difficult to achieve the optimal effect in practice. The recommendation system based on deep learning uses the ability of deep learning to extract hidden features and the successful application of deep learning in the field of computer vision and the improvement of computing power. For example, recurrent neural networks are often used to process timing information. In the recommendation system, it is used to integrate the current browsing history and browsing order and the time and context used to represent user behavior. By combining these representations with user preferences, Latent factors are combined to improve more accurate recommendations. In practice, user data is often very sparse, so that these models are challenging to achieve the most effective results in practice. The commonly used method to solve data sparsity and cold start introduce auxiliary information in the recommendation algorithm. As a semantic network, the knowledge graph can provide the recommendation system with semantically related information of users and items and give the users more exciting recommendations.

B. EXISTING METHODS, KEY CHALLENGES

Google's Thomas Mikolos et al. proposed Word2Vec [1] in 2013. This algorithm is network-based and maps a predictive text library to a matrix. In this matrix, each row is associated with a word in the input text data. For example, they found:

$$Vec\,(king) + Vec\,(\text{woman}) - Vec\,(\text{man}) \approx Vec(queen) \tag{1}$$

Researchers have discovered that there is much other information in the knowledge graph. While the translation model only uses structural information for representation learning, there is a lot of available information outside and in the actual knowledge graph. This information can improve the model's learning ability, thereby improving knowledge characteristics Express ability. Therefore, researchers have proposed to integrate this information to improve the quality of knowledge feature representation further. Literature [2] combined entity description information and proposed the DKRL model, which introduced the Freebase knowledge base to provide descriptive text information for entities. Two models represent the description semantics of entity pairs, the continuous bag of words model CBOW and the deep convolutional neural network model CNN. Literature [3] proposes to combine word2vec and TransE to complete knowledge feature representation. It mainly uses Baidu Encyclopedia, Wikipedia, etc., to extract information to form triples and then add these formed triples in the training word2vec. The head entity vector and The tail entity vector are closer, realizing the fusion of text knowledge and knowledge graph. PTransE [4] considers the multi-step relationship path and can also reflect the semantic relationship between entities. Whether it is from the original traditional model and a translation model that only uses triple structure information to a model that uses multi-source information, the purpose is to improve model learning in a certain way to improve the quality of representation.

Due to the increase in computing power and big data storage facilities, initially, literature [5] proposed a method of using a deep hierarchical model in the task of movie recommendation. Since this basic research, there have been several attempts to apply the deep model to recommender systems research. By leveraging the effectiveness of deep learning in extracting hidden features and relationships, researchers have proposed alternative solutions to recommendation challenges, including accuracy, sparsity, and cold-start issues. Literature [6] achieved higher accuracy by predicting the missing ratings of the user's item matrix with the help of autoencoders. At the same time, Devonholt and Bersini used neural networks to improve short-term prediction by converting CF into a sequence prediction problem. Forecast accuracy. Literature [7] proposed a deep model using CF to deal with the sparsity problem by learning a good representation. In addition, deep models are also used to deal with scalability issues, using the powerful capabilities of deep learning models in dimensionality reduction and feature extraction. By using deep neural networks to obtain low-dimensional features from high-dimensional features to propose a solution for scalability, the literature [8] uses deep learning to learn a deep-level nonlinear network structure, sign and further obtain the depth of user projects. Hierarchical feature representation and plays the role of dimensionality reduction. Deep learning extracts features from multiple heterogeneous networks, map different data to the same feature space, and obtains the exact vector representation.

The main body of the recommendation system based on the knowledge graph is the recommendation model, and there are many recommendation models. The common ones are traditional recommendation models, deep learning models, etc. The knowledge graph is used as auxiliary information to solve data sparsity and cold start of the recommendation system. At the same time, it brings diversity, accuracy, and interpretability to the recommendation system. CKE [9] introduces structural information, text data, image data, and other information in the knowledge base into the recommendation system to improve the quality of the recommendation system. Among them, the structure information uses TransR to obtain the vector features of the Entity, and the text data and image data use stacked denoising auto-encoding and stacked convolution auto-encoding to extract the vector features. Literature [10] proposed a content-based recommendation method that uses an extended activation algorithm on the DBpedia category structure to identify entities of interest to users. Literature [11] proposes a method of resuming music by calculating the semantic distance contained in the knowledge graph to make recommendations. Literature [12] proposed a graph-based cross-heterogeneous domain recommendation method. For each domain, a bipartite graph represents the relationship between its entities and features, and an effective propagation algorithm is designed to obtain Similarities between entities from different fields.

C. RESEARCH IDEAS, RESEARCH CONTENT

This paper researches two perspectives of the research on the combination of knowledge graph and recommendation system. The first is to combine the knowledge graph characteristics, study the knowledge feature learning, explore how better to learn the low-dimensional vector representation of entities and relationships, and provide high-quality auxiliary information for the recommendation system. The second is the recommendation technology research, studying how to make full and efficient use of the knowledge graph and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

combine the recommendation system with the knowledge graph. Based on comprehensive research and learning, corresponding innovation and improvement plans are proposed. The main research objectives and content of this article: (1) To improve the quality of the entity-relationship vector representation in the knowledge graph, this paper studies the knowledge representation learning model based on the self-attention mechanism and uses the characteristics of the semantic network of the knowledge graph to process the triples of the entity-relationship together, rather than separately. , To avoid the loss of information caused by the separate processing of the triples, combine the attention mechanism to use the difference of the importance of the triples to determine the semantic significance of the entity-relationship. (2) The historical behavior of users in the recommendation system is an essential source of recommendation information. This article studies the content recommendation model with the same embedded behavior and knowledge characteristics, studies how to mine user preference information from historical behaviors, and uses the self-attention mechanism to drill short-term or short-term users deeply—research how to effectively use the semantically related information of knowledge to improve the performance of recommendation. (3) Knowledge graph is a new technology. This paper studies a combination method to efficiently utilize the semantic network of knowledge graph and give full play to the features of knowledge graph knowledge semantic association network.

II. RELATED WORK

A. KNOWLEDGE FEATURE LEARNING ALGORITHM BASED ON TRANSLATION

(1)TransE

TransE [13] regards the relationship in the knowledge graph as a certain translation between the head entity and the tail entity. The relationship l_r can be regarded as the translation from the head entity l_h to the tail entity l_t. For each triple (h, r, t), TransE hopes to satisfy $l_h + l_r \approx l_t$, the model defines the loss function $f_r(h, t) = |l_h + l_r - l_t|_{L_1/L_2}$, which is the vector $l_h + L_1$ or L_2 distance between l_r and l_t. The TransE model has few simple parameters and low computational complexity, but it can directly establish complex semantic connections between entities and relationships. However, due to the simplicity of the TransE model, the effect is lacking when dealing with the complex relationships of large-scale knowledge graphs. According to the number of entities connected at both ends of the relationship in the knowledge graph, the relationship is defined as 1-1, 1-N, N-1, and NN. TransE has significant problems in handling these complex relationships. For example, the relationship r is N-1 relationship, the model will get $l_{h_0} \approx l_{h_1} \approx l_{h_2} \approx \cdots \approx l_{h_n}$, also when the relationship r is 1-N, it will also appear $l_{t_0} \approx l_{t_1} \approx l_{t_2} \approx \cdots \approx l_{t_n}$

(2)TransH In view of the deficiency of the TransH model in dealing with the complex relationships of 1-N, n-1, N-N, the paper [14] proposes that the TransH model uses different forms to represent the same entity in different relationships. The model thinks that for the same entity, it plays different roles in different relationships and should have different forms. The model uses both the translation vector d_r and the normal vector w_r of the hyperplane to represent the relation r. the head entity and tail entity vectors are mapped to the relation hyperplane along the normal vector w_r respectively, and then the translation operation is carried out, which is represented by $h_\perp$ and $t_\perp$as follows:

$$h_\perp = h - w_r^\top h w_r \tag{2}$$

$$t_\perp = t - w_r^\top t w_r \tag{3}$$

TransH is defined as the following scoring function:

$$f_r(h,t) = \| \ (h - w_r^\top h w_r) + d_r - (t - w_r^\top t w_r) \ \|_2^2 \tag{4}$$

(3)TransR Both TransE and Trans H assume that entities and relationships are embedded in the same space. However, an entity is a combination of multiple attributes, and different relationships focus on different Entity attributes. Intuitively, some similar entities should be close to each other in the entity space. But in the same way, in some specific and different aspects, they should be far from each other in the corresponding relationship space. In response to this problem, TransR [15] proposed to project entities and relationships into different spaces, that is, entity space and relationship space. For each triple (h, r, t), the TransR model first projects the Entity into the relational space through the matrix M_r and completes the translation in the corresponding relational space. The scoring function is as follows:

$$f_r(h,t) = \| \ M_r h + r - M_r t \ \|_{L_1/L_2} \tag{5}$$

A single relationship vector is not enough to establish all transitions from the head entity to the tail entity for the same relationship r, with multiple semantic representations. Literature [16] also proposed the CTransR model that the relationship can be divided in more detail. The CTransR model suggests different cluster entities and learns the representation of relationships in other clusters, which helps to improve the relationship between learning entities and relationships. The model first uses TransE to pre-train (h, r, t) triples, and clusters the connections, so that the head and tail entities will be classified into the corresponding clusters.

B. RECOMMENDATION ALGORITHM BASED ON DEEP LEARNING

Deep learning has been successfully applied in computer vision and natural language processing, and the reason for this success lies in the ability of deep learning models to extract features. Therefore, researchers have been triggered to introduce deep learning into the recommendation system and use deep feature extraction to improve recommendation performance.

RNN is specialized in processing timing information. In an e-commerce system, the user's current browsing history will affect their purchase behavior. However, most typical recommendation systems only create user preferences at the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

beginning of the session, which leads to ignoring the user's history and the sequence of user action sequences. The use of RNN in the recommendation system can integrate the current browsing history and browsing order to provide more accurate recommendations. RNN is also used to non-linearly represent the influence between the potential characteristics of users and items and their co-evolution over time. Literature [17] uses RNN to regard the recommendation problem as a sequence prediction problem and integrates the evolution of user tastes into the recommendation process. When analyzing the deep learning research of recommender systems, it is possible to get several results. Compared with the traditional nearest neighbor and matrix decomposition-based methods, RNN positively impacts the coverage of the following consumable product for recommendation and short-term prediction. This success stems from RNN's calculation of the evolution of user taste and the co-evolution of the potential characteristics of users and items.

Recommendation systems also benefit from CNN. When potential factors cannot be obtained from user feedback, CNN extracts elements from audio, text, and image data to generate user interest in recommendations. Literature [18] uses CNN to extract the latent features of the image and maps the features and user preferences to the same latent space. The semantic meaning of text information extracted using CNN is also used in recommendation systems, especially context-aware recommendation systems, to provide more qualitative recommendations. Therefore, for the recommendation, CNN is mainly used to extract potential factors and features from data, especially from images and text.

C. CONSTRUCTION OF KNOWLEDGE GRAPH

There are two ways to construct a knowledge graph: top-down and bottom-up. The bottom-up approach extracts entities, attributes, and relationships from data sources such as open databases or knowledge graphs. It then adds them to the data layer to induce and organize this knowledge to form a pattern layer. The top-down approach is the opposite. The technical architecture of the knowledge graph is shown in Figure 1:

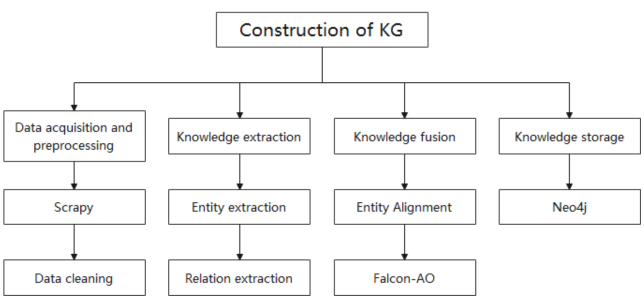

FIGURE 1. Knowledge Graph construction technology

The whole system is divided into four modules:

1) data acquisition and processing

The data of the system mainly includes structured data, semi-structured data, and unstructured data. The data sources are especially: Baidu Encyclopedia pages, various forums, public databases, and other data acquisition methods mainly use crawlers, and use the Scrap framework to build web crawler tools to obtain raw data. After obtaining the original data, it is preprocessed. The processing mainly includes: removing transfer characters in text data, removing stop words, removing repeated characters, and using regularization to disambiguate the format of each chapter name code.

2) knowledge extraction

This chapter will use the BiLSTM combined with the CRF method proposed in the literature [49]. Use a two-way long and short-term memory network combined with a conditional random field. BiLSTM extracts word features and CRF restricts the final entity tag to make it more reasonable. This model is one of the best performing models for named entity recognition tasks. The model structure is shown in the figure 2:

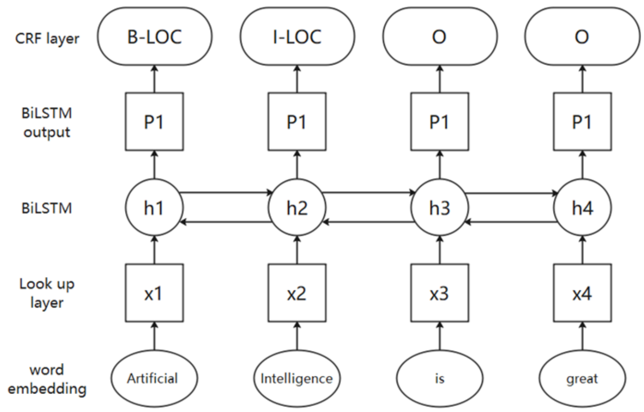

FIGURE 2. BiLSTM+CRF structure

3) knowledge fusion

Knowledge fusion realizes the elimination of ambiguity between reference items such as entities, relationships, attributes, and fact objects and then forms a high-quality knowledge graph. Knowledge fusion is merging two knowledge graphs. The fundamental problem is studying how to combine the description information of the same Entity or concept from multiple sources. The first thing that needs to be confirmed before is the types of knowledge: equivalent instances, equivalent classes, and equivalent attributes. This article will use the specific knowledge fusion tool Falcon-AO, an automatic ontology matching system, which has become a practical and popular choice for matching Web ontology expressed by RDF(S) and OWL.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

4) knowledge storage

This section uses a graph database neo4 for storage. Neo4j is a NoSQL database with a high-performance graph engine, which has all the characteristics of a mature database. It stores structured data on the network instead of in tables. Also, it is an embedded, disk-based, java persistence engine with complete transaction characteristics. In an attribute graph, a graph consists of vertices, edges, and attributes. Each vertex and edge can have one or more points. The graph created by Neo4j is a directed graph constructed with vertices and edges.

D. COMBINATION OF PRE-TRAINING MODEL AND KNOWLEDGE GRAPH

With the emergence of the pre-training model, the NLP direction ushered in a wave of revolution. The pre-training model has achieved amazing results in various tasks. With the emergence of various pre-training tasks in an endless stream, some researchers are considering how to introduce or strengthen the information contained in the knowledge graph into this type of model, thereby enhancing its ability to encode background knowledge or common sense information. ERNIE [20] mainly uses the high-information entity information proposed from the knowledge base, and enhances the corresponding representation in the text through a special semantic fusion module. K-BERT [21] mainly by modifying the attention mechanism in Transformer, and taking the relevant edges in the knowledge graph into account in the encoding process through a special mask method, the effect of the pre-training model is enhanced. KG-BART [22] presented aKG-augmented approach KG-BART based on pre-trained BART for generative commonsense reasoning. KGSynNet [23] tackle the task of entity synonyms discovery and exploit external knowledge graph and domain-specific corpus. resolved the OOV issue and semantic discrepancy in mention-entity pairs.

III. METHOD

A. KNOWLEDGE REPRESENTATION BASED LEARNING MODEL BASED ON SELF-ATTENTION MECHANISM

This chapter proposes a self-attention mechanism-based knowledge representation learning model KBSA, which learns the rich information implicit in the triples of entities and relationships through the attention mechanism, which improves the quality of knowledge feature representation. The self-attention-based knowledge representation learning model KBSA mainly considers their local semantic network characteristics. KBSA is primarily divided into two parts. One uses the self-attention mechanism to learn the inherently hidden complex and hidden information in the local triples around the Entity and integrate it. The second is that for relation representation, the self-attention mechanism is also used to extract semantic features from the triple context entity where the relation is located. At the same time, it is considered that different relationships in the knowledge graph pay other attention to entity semantics. We must learn the semantic representation of entities in a triple and the semantic representation of entity relationships in other triples for the semantics of entity relationships. The overall structure of the KBSA model is shown in the figure(Fig. 3).

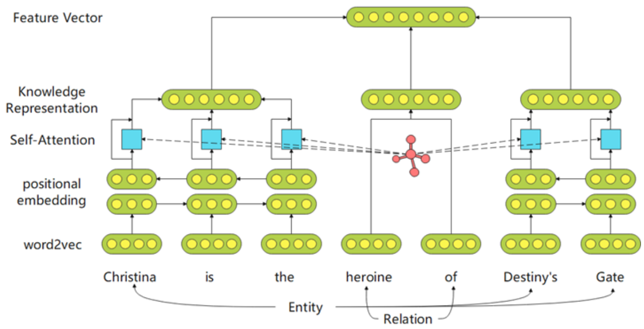

FIGURE 3. KBSA model

As illustrated in Figure 3, our model mainly consists of two parts. First, we build a neural network to embedding words and their positional information, then combine it with self-attention and knowledge graph. Then we synthesize knowledge representations into a feature vector.

B. CONTENT RECOMMENDATION MODEL WITH UNIFIED, EMBEDDED BEHAVIOR AND KNOWLEDGE CHARACTERISTICS

The content recommendation model with unified, embedded behavior and knowledge features tries to mine user preferences from knowledge networks and recommends users based on this. The model is mainly divided into three parts: historical preference learning, knowledge discovery, and candidate prediction. The overall framework is shown in Figure 3. The historical mining part digs out user interests from the user's historical browsing information. The knowledge graph part uses the semantic network information based on historical mining to dig user interests and hobbies further. At the same time, the knowledge representation learning method in Chapter 3 is used to represent the entities and relationships of the knowledge graph, respectively. The candidate prediction part uses the information learned by the knowledge graph part to predict the candidate recommended content. The RSHK model is a model that combines the user's historical data with the knowledge graph and is used in the recommendation system. The overall model mainly uses the attention mechanism. The overall structure is shown in Figure 4.

In the recommendation system, the user usually has a lot of click information, and the short-term or long-term interests of the user can be obtained from the click information. This part uses the self-attention mechanism to deeply explore the user's interests and hobbies through click behavior.

The input in this section is the user historical behavior matrix $H \in R^{M \times d}$, where M represents M historical user behaviors, and d is the historical behavior self-attention

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

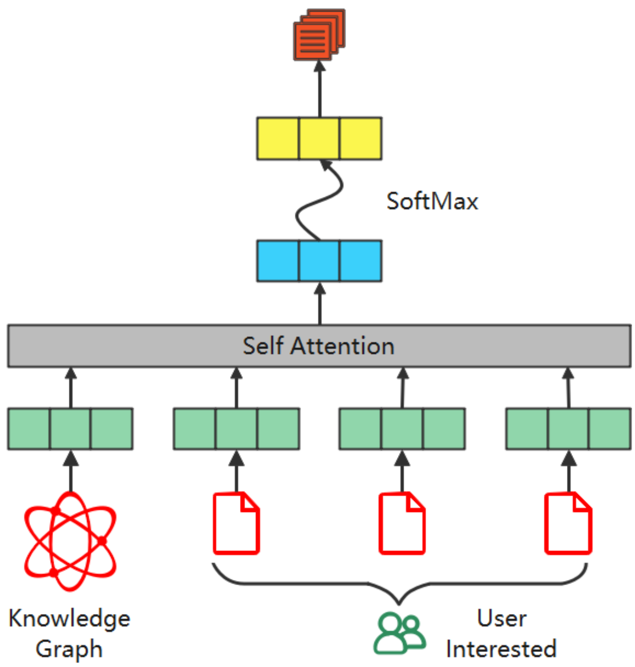

FIGURE 4. KBSA model

mechanism. The query, key, and value are all from the same source, and the historical behavior matrix H is used as the value, query, and key, obtained by nonlinear conversion.

$$Q = ReLU(HW_Q) \tag{6}$$

$$K = ReLU(HW_K) \tag{7}$$

Among them, $W_Q \in R^{d\times d} = W_K \in R^{d\times d}$ are the weight matrices of query and key respectively, and ReLU is an activation function. The weight matrix is calculated as follows:

$$S = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \tag{8}$$

The output is a similarity matrix of M's historical behavior. Finally, the similarity matrix and value are multiplied to get the output of self-attention.

$$\alpha = SH \tag{9}$$

$\alpha \in R^{L\times d}$ can be regarded as the expression of the user's interest. This chapter calculates the mean value of self-attention expressions to represent the user's interests and hobbies. Where m is the user preference vector.

$$m = \frac{1}{L}\sum_{l=1}^{L}\alpha_l \tag{10}$$

C. TITLE

IV. EXPERIMENT

A. KNOWLEDGE REPRESENTATION LEARNING MODEL EXPERIMENT

The three experiments designed in this chapter show the model of this chapter from different angles.

(1) The first type of link prediction experimentally predicts the missing triple's head or tail entity.
(2) The second experiment classifies triples classifies correct triples and wrong triples.

The experiment is carried out on the open data sets, and the investigation is compared with other knowledge representation learning methods. Perform experiments on the data sets to evaluate the model and analyze the experimental results. The results of selected classic models such as TransE, TransH, TransR, TransD, and TransA under relevant practical tasks will be compared. A detailed introduction is given in the second section of this chapter, and the first section introduces the experimental environment and the data sets used in the experiment. The results of the most classic and earliest, and widely used TransE in the translation series are used as the experimental baseline values. These two tasks evaluate the ability to predict unknown triples from different perspectives and application environments and assess feature representation quality.

The experimental environment is shown in Table 1.

TABLE 1. Experimental environment

Environment	Parameter
Processor	12cores
Memory	16G
Graphical card	GeForce GTX 2070 8GB
Architecture	TensorFlow2.0

The experiment uses four data sets, Wordnet subsets WN18 and WN111, Freebase subsets FB15k and FB13. FB13 contains 75043 entities and 13 relationships, FB15k contains 14,951 entities and 1345 relationships, WN11 has 38696 entities and 11 relationships, and WN18 has 40943 entities and 18 relationships. In contrast, FB15k contains more entity relationships and triples. The specific data information is shown in Table 2.

TABLE 2. Data sets situation

Data sets	Entity	Relation	Triples		
FB13	14783	24	Train	Valid	Test
FB15k	41704	2134	124451	24114	324245
WN11	85235	41	235345	4542	214141
WN18	25462	121	147654	13455	23256

1) Link prediction

Experimental method: This experiment is used to predict the link of triples. The task is to correctly fill the missing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

triples of the head entity h or tail entity t, that is, predict the missing head entity h or tail of the correct triple (h, r, t) Entity t. For triples (h, r, t), the tail entity t is predicted given the head entity and the relationship (h, r), and the head entity h is expected given the relationship and the tail entity (r, t). In fact, in experimental evaluation, the purpose of this task is not to find the best answer entity but to place more emphasis on sorting a set of candidate entities from the knowledge graph. Following other knowledge representation methods, this experiment uses the WN18 and FB15k data sets used in the translation series model as the link prediction experiment's data sets.

Evaluation indicators: The following two indicators are used:

(1) Mean Rank represents the average rank of the correct head entity or tail entity of the negative triple

(2) HITS@10 indicates the proportion of triples in the top 10 correct entities to all triples.

This experiment follows the same evaluation method as the translation series model. For each triple (h, r, t) in the test set, replace the head entity h randomly to obtain the negative triple (h', r, t), and calculate the scoring function f for all negative triples (h, r, t) score, and finally sort the scores of all negative triples and positive triples in ascending order, and record the ranking of the original correct triples according to the sorting results. In the same way, the tail t forms a negative triple (h, r, t') and gets the rank of the original correct triple. In the experiment, record the proper head or tail entity of each negative triple in the ranking of candidate entities. Due to the evaluation indicator MEAN, if the correct Entity is in the top ten rankings, it is called a hit and recorded for HITS@ 10 reviews

At the same time, you need to consider whether it is still the correct triple after replacement. The relationship in the knowledge graph is usually complex and multi-type, such as one-to-many, many-to-one, and many-to-many. In this case, the negative triples constructed by random replacement may still be the correct triples. Groups, even such "negative" triples, have higher scores than the original correct triples, which will affect the experiment and cannot effectively demonstrate the model learning ability of this chapter. To ensure that the generated negative triples are entirely wrong and avoid involving the experiment. The experiment will eliminate these negative triples that are correct when constructing negative triples. That is to say, such triples appearing in the training set, validation set, or test set are deleted, and the experiment is set to "Filt", on the contrary, the experiment that did not delete the operation is set to "Raw".

Experimental parameters: This article uses a small batch stochastic gradient descent algorithm to train the model. The experimental parameters mainly include vector dimension d, learning rate μ, interval value γ, and balance hyper parameter input. The vector dimension d of entities and relationships is selected in 20, 50, 80, 100, the learning rate μ is selected in 0.01, 0.005, 0.001, and the interval value γ is selected in 0.5, 1.0, 2.0, 3.0, The batch size value B is selected in 100, 500, 1000, 2000. The final optimal parameter configuration will be determined on the verification set. The specific parameters are shown in the Table 3.

TABLE 3. Link prediction experiment parameters

Data sets	Embedded dimensions	Learning rate	Interval value	Batch
WN18	80	0.01	2.0	500
FB15k	100	0.001	2.0	1000

Experimental results and analysis: The experimental results of standard link prediction are shown in Table 4. The experiment was carried out under two experimental conditions of FB15k and WN18 data sets. Under the condition of "Filt", MEAN and HITS@10 are better than the "Raw" condition. The original translation model TransE has poor model performance due to its simple model. In the comparison model, TransA and TranSparse perform relatively better. The model proposed in this paper performs slightly better than the superior Trans A on the two data sets. Compared with the TranSparse model, and better than the basic model TransR, TransR learns the semantic information of the entity-relationship from the perspective that the entity-relationship should be in a different space. Still, it ignores the surrounding triples' implicit information when learning the triples' structure information. The model proposed in this article is aimed at this problem. It retains the information contained in the triples around the Entity through the attention mechanism to avoid losing information when learning the triples.

According to experimental results, there are still some differences in the performance of all models on the two data sets. The MEAN test result on FB15k is lower than that on WN18, indicating that the correct Entity on FB15k ranks higher and the overall correct Entity on WN18. The ranking is more inferior. The test results of HITS@10 are significantly lower than WN18 on FB15k, and HITS@10 is generally around 90 under the "Filt" condition on WN18. The performance structure is reasonable, indicating that the model's predictive ability on WN18 is more vital. The experimental results of the model proposed in this article are very close to TransA and TranSparse. By analyzing the specific conditions of the two data sets, one of the reasons for this is that FB15k has a more extensive data volume, which has more entities and ternary than WN18. In the case of insufficient training, the KBSA model failed to use the model to effectively learn the confidential information of the triples around the Entity. Although the amount of data in WN18 is less than that of FB15k, a small number of triples makes the semantics of the entity-relationship The embodied is more specific and detailed. But combining the experimental results on the two data sets, the model proposed in this paper still effectively learns the rich information hidden by all triples of the Entity to a certain extent and avoids the separate processing of the triples—loss of information studying.

The above-mentioned ordinary link prediction does not

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

TABLE 4. Link prediction experiment results

Datasets	FB15k				WN18			
Metric	MEAN		HITS@10 (%)		MEAN		HITSQ10 (%)	
	Raw	Filt	Raw	Filt	Raw	Fil	Raw	Fil
TransE	253	135	36.9	48.1	257	248	71.4	83.2
TransH	204	81	41.5	57.5	314	301	70.9	82.7
TransR	216	76	42.8	64.5	237	209	72.3	85.7
TransD	221	77	48.4	73.2	248	223	73.2	86.5
TransA	165	78	55.1	79.4	305	262	76.3	87.3
TranSparse	215	68	50.9	52.7	233	231	73.6	87.4
KBSA	208	71	42.7	80.6	223	216	83.2	89.6

TABLE 5. Experimental results of relationship types on FB15K

Task	Predicting left(HITS@10)				Predicting right(HITS@10)			
Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
Unstructured	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(Linear)	35.1	53.7	9.0	40.3	32.7	14.9	61.6	43.3
SME(Bilinear)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
KBSA	69.8	84.3	29.7	66.5	65.3	40.1	86.6	69.4

consider the complex type of the relationship and treats the relationship as the same. However, in the actual knowledge graph, the relationship type is usually more complicated, such as 1 to N, N to 1, etc. This article carries out link prediction for complex relationships and divides link prediction into head entity prediction, left, and tail entity prediction, right. Table 6 shows the link prediction experiments for different relationship types. From the experimental test results, the KBSA model is significantly better than other comparable models. These results show that the KBSA model is also potent in the face of complex relationships.

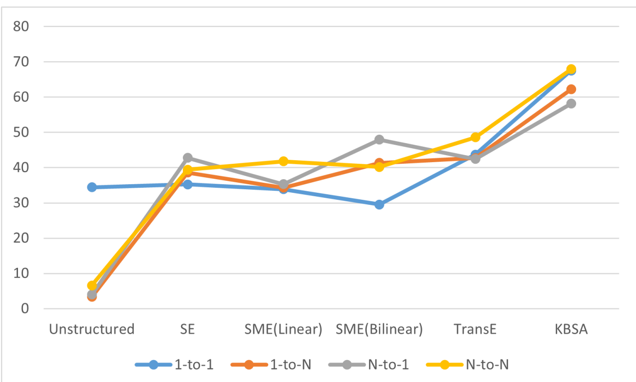

FIGURE 5. Average results of relationship types model

2) Triple classification

Experimental method: Triple classification is to classify correct and wrong triples, regarded as a binary classification task. The data sets used in the experiment are three data sets: WN11 and FB13 data sets provided by literature [19], and FB15k data sets. The WN11 and FB13 data sets are triples marked, the wrong triples are marked as -1, and the correct triples are marked as 1. However, the two data sets, WN11 and FB13, contain fewer relationships, so that this experiment will add the FB15k data sets to the test. The translation series model also uses the FB15k data sets to classify triples. Still, the triples are not labeled in FB15k, so negative samples and labels need to be constructed before the experiment. This paper uses random replacement to build the same number of error triples as the correct triples, randomly select entities from the entities connected by the relationship of the triples to replace the head entities of the correct triples to form the error triples. This method can avoid the obvious unrelated triples in the wrong triples. The replaced triples may also be the correct triples, resulting in an imbalance between the positive sample and the negative sample. So correct triples should be eliminated negative piece.

Evaluation indicators: The evaluation method of the triad classification task is to classify according to the score function of each triad, in which a triad score threshold is set for each relationship. Suppose the triad score is higher than the threshold of the corresponding relationship. In that case, it will be considered a correct triple, and the reverse is below the point. It is regarded as a wrong triple. The corresponding score threshold is obtained through training, and the final value is obtained by following the model's verification on the validation data sets. The accuracy rate is used as the evaluation index, and the number of correctly classified triples is divided by the number of all triples. The formula is shown in formula 11

$$Precision = \frac{Correction}{All} \times 100 \tag{11}$$

Experimental parameters: The parameter selection method is the same as the link prediction obtained in the verification set. For all data sets, 800 iterations of training are performed in this article. The best parameter configuration will be determined on the verification set based on the verification effect. The specific parameters are shown in Table 6.

TABLE 6. Parameter table

data sets	Embedded dimension	Learning rate	Interval value	batch size
WN11	50	0.01	3	I 000
FB13	50	0.01	2	500
FB15k	100	0.001	2	1000

The results of triple classification are shown in Table 7 and Figure 6. Overall, the KBSA results of the model presented in this paper are higher than those of the baseline model TransE and better than the base model TransR. The classification accuracy of FB15k is better in the three experimental data sets, which may be due to the more significant number of experimental ternaries in FB15k than in WN11 and FB13. The results of the KBSA model on WN11 and FB13 data sets are close to those of TransA and TranSparse models with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

TABLE 7. Experimental analysis

Data sets	WN11 (%)	FB13 (%)	FB15k (%)
TransE	72.1369	67.8724	78.9764
TransH	76.6041	78.3239	71.854
TransR	81.7823	73.1126	78.0505
TransD	85.3061	82.8631	84.2176
TransA	82.5111	85.1076	87.7639
TranSparse	84.7952	84.1251	85.9356
KBSA	87.2309	87.4153	90.0272

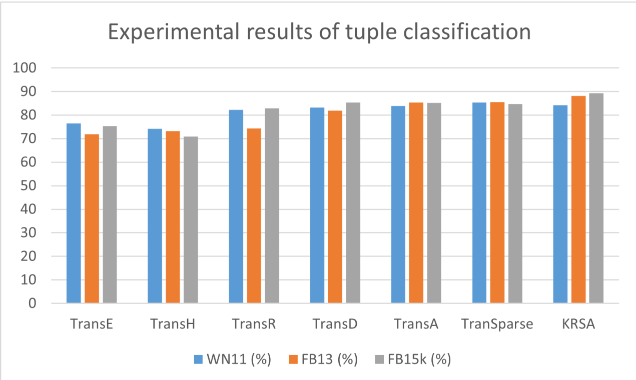

FIGURE 6. Tuple classification

good experimental results but significantly better than FB15k. The work shows that the KBSA model can learn the semantics of entities and relationships from triplets and improve the quality of knowledge representation when facing large-scale knowledge maps. The overall results fully validate this chapter's idea, using the self-attention mechanism to learn entity information in all triples, and avoiding the loss of information caused by processing the triples separately.

B. CONTENT RECOMMENDATION MODEL EXPERIMENT UNIFIED EMBEDDED BEHAVIOR AND KNOWLEDGE CHARACTERISTICS

This chapter uses the following three data sets in the experiment: movies, books, and news. MovieLens-1M6 is a widely used benchmark data sets in movie recommendation, which contains approximately 1 million score data on the MovieLens website. The Goodbooks-10k data sets contain 1,149,780 scoring data for Book-Crossin communities. The Google news data sets contain 1025192 hit news items collected from Bing News's server logs from October 16 to August 8, 2016. Each news item consists of a headline and a paragraph of content. MovieLens-1M and Goodbooks-10k are not marked. Each item is marked as 1 in this chapter, indicating he has clicked on the item marked as 0 for each setting user has not clicked. This chapter uses Microsoft Satori to construct a knowledge graph for each data set.

TABLE 8. Specific circumstances of the data sets

Data sets	MovieLens 100K	Goodbooks-10k	Google news
Users	6036	17860	141487
Items	2445	14746	535145
Interactions	753772	139746	1025192
1 -hop triples	20782	19876	503112
2-hop triples	178049	65360	1748562
3-hop triples	318266	84299	3997736
4-hop triples	923718	71628	6322548

In REKC, this chapter sets the number of jump ranges H=2 for MovieLens-1M and Goodbooks-10k, and H=3 for Google news. Experimental results show that a more significant number of H values can help improve performance and increase calculation. This method has specific practical value. On the MovieLens-1M data sets, the embedding dimension d of the project and knowledge graph is 16, and the learning rate η is 0.01. The dimension d of Goodbooks-10k is 4, and the learning rate η is 0.001; the dimension d of Google news is 32, and the learning rate η is 0.005. AUC determines the final optimal parameters on the validation set.

For each data set, the ratio of the training set, validation set, and test set is 6:2:2. Each experiment was repeated ten times, and the average value was taken for performance analysis. This chapter uses click-through rate (CTR) prediction experiments to verify the model proposed and applies the trained model to each interactive output in the test set to predict the click probability. Use Accuracy and AUC to evaluate the effect of CTR prediction.

The experimental results are shown in Table 9, Figure 4-3, 4-4, 4-5:

TABLE 9. Experimental results of AUC and Acuracy on CTR prediction

Model	MovieLens 100K		Goodbooks-10k		Google news	
	AUG	ACC (100%)	AUG	ACC (100%)	AUG	ACC (100%)
CKE	0.796	0.739	0.674	0.635	0.56	0.157
SHINE	0.778	0.732	0.668	0.631	0.554	0.537
DKE	0.655	0.589	0.621	0.598	0.661	0.604
PER	0.712	0.667	0.623	0.588	0.636	0.677
LibFM	0.892	0.812	0.685	0.639	0.644	0.588
Wide&Deep	0.903	0.822	0.711	0.623	0.654	0.595
REKC	0.931	0.846	0.735	0.663	0.694	0.622

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3106914, IEEE Access

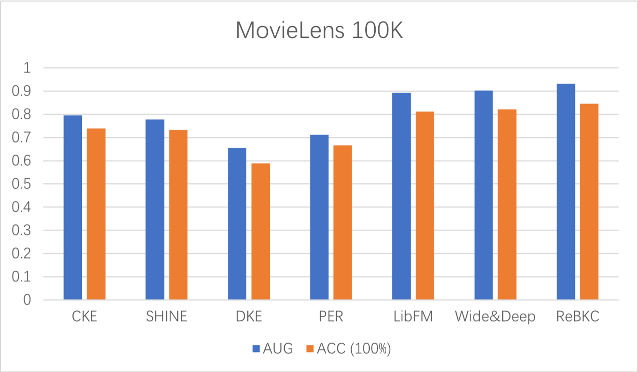

FIGURE 7. MovieLens 100K

FIGURE 8. Goodbooks-10k

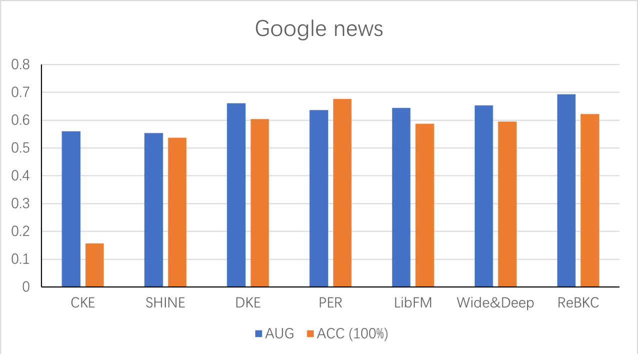

FIGURE 9. Google news

CKE performs relatively poorly in all models. CKE combines text, visual, and structured knowledge into the recommended model, but only structured knowledge is used here.

Multiple comparison models, SHINE, perform better in movie and book recommendations than in the news. Because there are more entities in the information, more negative noise signals are introduced, and it is too complex to input a triple corresponding to one jump of an entity. The DKN model, on the other hand, performs best in news recommendation but performs slightly worse in movie and Book recommendation. The DKN model was designed to extract text and knowledge map entity features through a convolution neural network for news complexity. Movie and book names are usually short and vague, making it difficult to extract useful information. In contrast, news generally exists in the form of text and provides helpful information for models.

PER does not perform well in movie and Book referrals because user-defined meta paths are hardly optimal.

In addition, due to the complexity of entity types and relationships involved in news, meta paths cannot be predefined and, therefore, cannot be applied to news recommendations. LibFM and Wide&Deep have achieved satisfactory performance as two general recommendation tools because they extract user, project, and knowledge entity characteristics in-depth in different ways.

The REKC model in this paper performs well on the MovieLens-100K and Goodbooks-10k models. The performance on the Google news data sets is not as good as the first two data sets, but it is also better than other comparable models. The lower performance on the Google news data sets maybe that news is more complex. There are many entities in the information which continuous appearance of some entities brings noise to the model. On the whole, this shows that the REKC model mines users' short-term or long-term preferences from historical behaviors, and adds them to the knowledge graph, and uses the attention mechanism to further dig out user preferences in the knowledge graph, filter information that users are not interested in, and fully discover The user's selection can effectively improve the recommendation ability.

V. CONCLUSION

This paper presents a learning model of knowledge representation based on the self-attention mechanism. In this paper, using the graph structure of knowledge map, the self-attention mechanism is used to represent entity learning features, the self-attention tool is used to handle complex relationship feature representation learning, which solves the problem of insufficient expression of semantic features caused by traditional translation models and modeling combined with multi-source information.

A content recommendation model based on the unified embedding of behavior and knowledge features is presented. This paper uses a self-attention mechanism to mine users' short-term or long-term preferences from each user's log history behavior. It combines historical choices with knowledge graph semantic networks to further explore users' preferences. To solve the shortcomings of the current recommendation model based on a knowledge graph that can not effectively utilize the semantic network information of the knowledge graph.

Experiments have proved the validity of self-attention-based learning methods for knowledge representation and recommendation methods combining historical behavior sequence and knowledge map. Especially in learning knowledge characteristics, sufficient experimental validation has been carried out to prove the feasibility of learning knowledge semantics by covering the whole triple. As well as in the combination of knowledge map and recommendation algorithm, click prediction experiments demonstrate that it is feasible to use user history to learn user preferences and combine knowledge map to different mine user preferences.

REFERENCES

[1] Church K W. Word2Vec[J]. Natural Language Engineering, 2017, 23(1): 155-162.
[2] Xie R, Liu Z, Jia J, et al. Representation learning of knowledge graphs with entity descriptions[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2016, 30(1).
[3] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
[4] Hu Xiaoxiang. Analysis and Thinking on Medical Advertising Chaos [J]. Doctor Online, 2018, 8 (12): 44-45.
[5] Yuan Kaiqi, Deng Yang, Chen Daoyuan, Zhang Bing, Lei Kai.Construction technology and research progress of medical knowledge graph [J]. Computer Application Research, 2018, 35 (07): 1929-1936.
[6] Ruan Tong, Sun Chenglin, Wang Haofen, et al. construction and application of knowledge graph of traditional Chinese medicine [J]. Journal of Medical Informatics, 2016, 37 (4): 8-13.
[7] Hou Mengwei, Wei Rong, Lu Liang, et al. Review of knowledge graphing research and its application in the medical field [J]. Computer Research and Development, 2018, 55 (12): 2587.
[8] Berant J, Chou A, Frostig R, et al. Semantic parsing on freedom from question-answer pairs [C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2013: 1533-1544.
[9] Yao X, Durme B V. Information Extraction over Structured Data: Question Answering with Freebase [C]//Meeting of the Association for Computational Linguistics. 2014.
[10] Bordes A, Chopra S, Weston J. Question Answering with Subgraph Embeddings [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 615-620.
[11] LIU Bin, CHEN Hua. Discussion on Vector Space Model Information Retrieval Technology [J]. Intelligence Journal, 2016 25 (007): 92-93.
[12] Wang Q, Liu J, Luo Y, et al. Knowledge base completion via coupled path ranking [C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016: 1308-1318.
[13] Lee Tae Seok, Shin Su Mi, Kang Seung Shik. Named Entity Recognition for Patent Documents Based on Conditional Random Fields [J]. KIPS Transactions on Software and Data Engineering, 2016, 5 (9).
[14] Lesk M. Automatic sense disarmament using machine-readable dictionaries: how to tell a pine cone from an ice cream cone [C]//Proceedings of the 5th annual international conference on Systems documentation. 1986: 24-26.
[15] Yang Xinxin, Wang Yingshuai. Design and implementation of a personal name disambiguation system based on improved K-means algorithm [C]//7th National Conference on Web Information Systems and Their Applications, 5th National Conference on Semantic Web and Ontology, 4th National Conference on Electronic Government Technology and Applications. China Computer Federation, 2012: 10-12, 17.
[16] Niwattanakul S, Singthongchai J, Naenudorn E, et al. Using of Jaccard coefficients for keywords similarity [C]//Proceedings of the international multicultural of engineers and computer scientists. 2013, 1 (6): 380-384.
[17] Wang X, Pao D. Memory-based architecture for multicharacter Aho-Corasick string matching [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017, 26 (1): 143-154.
[18] Bag S, Kumar S K, Tiwari M K. An effective recommendation generation using relevant Jaccard similarity [J]. Information Sciences, 2019, 483: 53-64.
[19] Bergmanis T, Goldwater S. Context-sensitive neural lamentation with elements [C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Languages: Human Language Technologies, Volume 1 (Long Papers). 2018: 1391-1400.
[20] Zhang Z, Han X, Liu Z, et al. ERNIE: Enhanced Language Representation with Informative Entities[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 1441-1451.
[21] Liu W, Zhou P, Zhao Z, et al. K-bert: Enabling language representation with knowledge graph[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(03): 2901-2908.
[22] Liu Y, Wan Y, He L, et al. KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning[J]. AAAI2021.
[23] Yang Y, Yin X, Yang H, et al. KGSynNet: A Novel Entity Synonyms Discovery Framework with Knowledge Graph[J]. DASFFA2021.

ZHIXUE JIANG was born in Taiyuan, Shanxi Province, China, in 1996. He received the B.S. in University of Science and Technology Liaoning, and he is currently pursuing the M.S. Degree in there. His mentor's main research direction is Search Engine, automatic abstract, question and answer system, etc.

• • •