Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Enhanced Language Model with Hybrid Knowledge Graph for Mathematical Topic Prediction

Canghong Jin[1] | Wenkang Hu[2] | Yabo Chen[2] | Minghui Wu*[1]

[1]College of Computer Science and Technology, Zhejiang University City College, Hangzhou 310015, China
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou 310012, China

Correspondence
Minghui Wu, Zhejiang University City College, Hangzhou, China
Email: mhwu@zucc.edu.cn

Abstract

Understanding mathematical topics is important for both educators and students to capture latent concepts of questions, evaluate study performance, and recommend content in online learning systems. Compared to traditional text classification, mathematical topic classification has several main challenges: (1) the length of mathematical questions is relatively short; (2) there are various representations of the same mathematical concept(i.e., calculations and application); (3) the content of question is complex including algebra, geometry, and calculus. In order to overcome these problems, we propose a framework that combines content tokens and mathematical knowledge concepts in whole procedures. We embed entities from mathematics knowledge graphs, integrate entities into tokens in a masked language model, set up semantic similarity-based tasks for next-sentence prediction, and fuse knowledge vectors and token vectors during the fine-tuning procedure. We also build a Chinese mathematical topic prediction dataset consisting of more than 70,000 mathematical questions with topics. Our experiments using real data demonstrate that our knowledge graph-based mathematical topic prediction model outperforms other state-of-the-art methods.

KEYWORDS:
mathematical topic prediction; knowledge graph; language fusion mode; intelligent education

1 | INTRODUCTION

How best to teach conceptual and procedural knowledge in mathematics is an open question in education. Procedural knowledge is defined as "Learning that involves only memorizing operations with no understanding of underlying meanings", whereas conceptual knowledge is "Explicit or implicit understanding of the principles that govern a domain and of the interrelations between pieces of knowledge in a domain"[1]. Given certain knowledge, it is possible to design procedural knowledge-based or conceptual knowledge-based questions. Therefore, in terms of teaching and learning, knowledge points have several uses, including developing auto-generated test systems, measuring the study abilities of students, and influencing the practice-based theory of mathematical knowledge for teaching.

Correctly predicting the knowledge point to which a question belongs is not a trivial task. There are three main challenges, as follows. (1) **Short context classification**: for given mathematical questions, the length of the context is usually shorter than the original text of the classification task, in our case, the average length of question text (calculate the length of different tags). How to learn using such short texts, especially in the pre-training procedure, is an important problem. (2) **Mathematical knowledge**

point encoding: the entities and relationships in the mathematical knowledge graph should be extracted and encoded in the question classification task using a language model. (3) **Heterogeneous information learning**: a mathematical question contains both normal content and mathematical keywords, resulting in two individual vector spaces. Distinguishing lexical, syntactic, and keywords related to mathematics in a question from the normal context is a difficult task, as is fusing these two different content types in the pre-training and fine-tuning procedures.

Mathematical symbols are not just abbreviations, and students need to learn to understand each symbol in the context of a variety of concrete situations, pictures, and languages. The symbols of mathematics allow us to both discover and express relationships between concepts.

Example 1: There are 100 orange trees in an orchard, and each tree has an average of 600 oranges. Several orange trees can increase their yield, but if this occurs, the distance between the trees and the sunlight received by each tree will decrease. According to experience, each tree will produce an average of five oranges. Find the functional relationship between the number of trees grown and the total yield of oranges.

Example 2: How many real roots does the following equation have? $x^2 + 3x + 4 = 0$.

To overcome the challenges mentioned above, we propose a new method called $KG\text{-}MTP$ (knowledge graph-based mathematical topic prediction), which pre-trains a mathematics representation model using both large-scale tagged questions and knowledge graphs for use in mathematical education.

Motivated instances: There are three types of question: (1) short content; (2) long content with knowledge entities; (3) mathematics world problems. Although they have different representations in terms of content, they have the same topic.

To facilitate the study and evaluation of mathematical topic prediction tasks, we built a novel benchmark dataset named Chn-math, based on middle school mathematics. For most mathematical concepts, teachers design different content and use different symbols and questions to evaluate students' understanding. Therefore, we directly collected real questions used in middle school education to build a benchmark. Next, we invited several mathematics teachers to provide a question on each of several (at most five) topics. Moreover, in order to improve the quality of the dataset, we removed those items that contained fewer than 10 words. We call this benchmark dataset Chn-Math.

The main contributions of this paper are as follows.

- We propose an pre-trained method that uses word and entity encoding to predict mathematical topics. Our model is based on $BERT$ but integrates a mathematical knowledge graph. We also propose some novel tasks during pre-training processing to improve the performance of the model. Our method can be applied not only to mathematical questions but to all classification problems.
- We introduce a novel topic prediction task for mathematical questions in Chinese. Mathematical questions contain normal text, mathematical symbols, pictures, and concrete instances, in contrast to other short texts. This represents an interesting and significant contribution to NLP(natural language processing) research.
- We evaluate the accuracy of the top k items in the Chn-Math dataset. Compared with state-of-the-art deep learning models, such as $BERT$, the evaluation results demonstrate that our model outperforms other baselines.

The structure of this paper is as follows. In Section 2, we discuss related work in two categories: text classification methods and natural language models. In Section 3, we present our motivation and notations used in our problems. Section 4 shows the architecture of our model($KG\text{-}MTP$). In Section 5, we present experimental results obtained using our framework. Section 6 present our conclusion and several directions for future work.

2 | RELATED WORK

Mathematical question topic prediction is closely related to other research areas, including text classification and natural language processing, which focus on programming computers to process and analyze large amounts of natural language data. In this section, we provide a brief review of related works in two categories: text classification methods and natural language models.

2.1 | Text Classification Method

Text classification is a similar problem to mathematical topic prediction, especially in the case of short texts. The task in text classification is to assign a document to one or more classes based on its content. Short texts have natural characteristics including

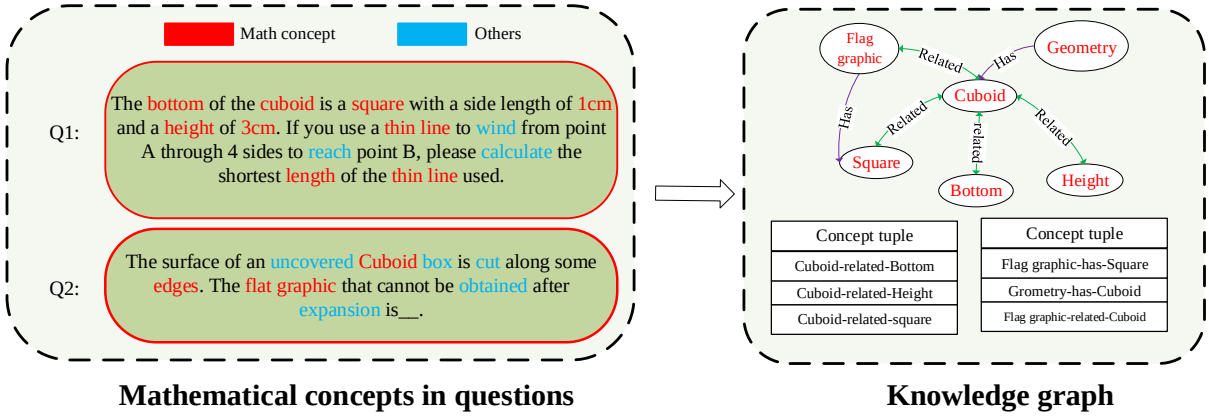


FIGURE 1 Model learning from a new dataset to generate novel nodes and edges

sparseness, large scale, immediacy, and non-standardization[2]. Traditional short-text classifying methods leverage semantic and topic models (e.g, latent Dirichlet allocation) with single or multiple levels of granularity[3]. In recent years, text classifiers have been designed using deep learning models to improve performance; these include word embedding-based methods such as Pte[4] and LEAM[5], and graph convolutional network-based methods such as SSC-GCN[6] and Text GCN[7].

Our work uses similar ideas to those of these methods, the major difference is that we use a pre-trained model with a knowledge structure instead of word embedding or structure embedding.

2.2 | Pre-Trained Natural Language Model

There are many pre-trained language representation models for capturing information from text in various NLP tasks. Pre-trained models can train auto-encoders on an unlabeled corpus with fine-tuning for special tasks. The best of these methods is the deep bidirectional model with multiple layer transformers ($BERT$) proposed in 2018[8]. XLNet integrates an auto-regressive model in the pre-training process and outperformed $BERT$ on 20 NLP tasks[9]. Other methods optimize encoding methods in the pre-training procedure and also show better performance than $BERT$; these include SpanBert[10], $ERNIE$ ($BERT$ with knowledge graphs)[11], and $ERNIE$ (Baidu)[12].

Our work refers to ideas and architectures from the above-mentioned pre-training techniques; however, unlike these models, our model integrates a knowledge graph and proposes novel masking and prediction strategies to enhance entity representation.

3 | PRELIMINARIES

3.1 | Mathematical Understanding and Motivation

Conceptual understanding involves knowing more than isolated facts and methods. The successful student understands mathematical ideas and has the ability to transfer their knowledge into new situations and apply it to new contexts.

There are four key components of mathematics questions: concrete experiences, symbols, language, and pictures. Here, we focus on first three components. The connections between symbols, language, and concrete experiences can be developed and established to improve students' mathematical understanding. Our motivation is how to distinguish these components in questions and extract latent semantics during the pre-training and fine-tuning procedures. As shown in Figure 1, we use knowledge graphs to help the model to navigate and to enhance use and understanding of the mathematics corpus.

3.2 | Problem Formulation

In this section, we first define several basic concepts and then provide a formal definition of knowledge point classification problems. Detailed descriptions of the notation used in our problems can be found in Table 1.

TABLE 1 Notation used in our problems

Notation	Description
q	Mathematical question
e	Knowledge entity; e_q is the entity in question q
r	Knowledge entity relations; $r(e_i, e_j)$ is the relation between entities e_i and e_j
$\mathcal{KG}$	Knowledge graph, $\mathcal{KG} = \{e, r\}$
$\mathcal{T}$	Topic set; t is the concrete topic and $t \subset \mathcal{T}$
t_i^q	i-th topic of q

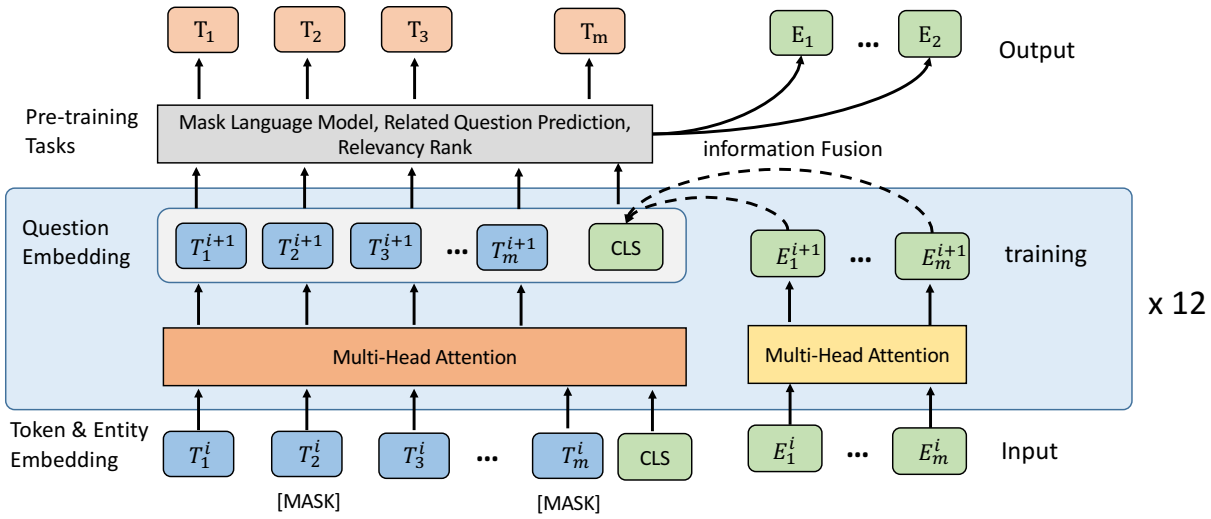


FIGURE 2 Architecture detail of $KG\text{-}MTP$

The **mathematics topic prediction** problem refers to how to judge, given a question q and total topics $\mathcal{T}$, whether each topic t belongs to q.

4 | METHODOLOGY

In this section, we describe the construction of our model, including (1) the overall framework architecture of $KG\text{-}MTP$; (2) encoding of knowledge entities in knowledge graphs; (3) two pre-training procedures: masking tokens by their mathematical semantics, and predicting the next question via related knowledge entities; and (4) details of the fine-tuning procedure with token and mathematical semantic fusion in questions.

4.1 | Model Architecture

The architecture is shown in figure 2.The architecture of the $KG\text{-}MTP$ model consists of three main separate procedures: (1) generating the underlying textual encoder to capture basic lexical and external mathematical information in a knowledge graph; and (2) setting up three tasks: an advanced mask content model with knowledge entities, mathematical concept similarity-based next-question prediction, token and knowledge entity fusion in the transfer procedure, and details of fine tuning with entity semantics.

4.2 | Knowledge Entity Embedding

Based on mathematical education concepts, we need to embed components of the knowledge graph, including entities and relations, into continuous vector spaces[13]. The mathematical knowledge graph contains three different relations between two entities: *subclass*, *has*, and *is*. For simplicity, we use a translation distance model TransE[14] to exploit the relations as vectors in the same space $\mathbb{R}^d$. Given two embedded entities e_i and e_j connected by r, $e_i + r \approx e_j$ when (h, r, t) holds. Other TransE extend methods could also be implemented in our knowledge graphs, such as TransH[15], TransR[16], or KG2E[17].

4.3 | Pre-Training with Knowledge Graphs

Semantic Mask Procedure Although the masked language model (MLM) is strictly more powerful than a bidirectional model or single directional model, randomly selected WordPiece tokens in the MLM procedure would cause loss of the semantics of mathematical questions. Similar to the use of MLM in $BERT$ pre-training, here, we select masked tokens with their related knowledge entities. We refer to this procedure as "semantic MLM" (SMLM). In all of our experiments, we set a parameter δ to determine whether to mask a normal token or knowledge entity. For each sentence in a mathematical question, we replace the i-th token with a [MASK] token if the generated random value is larger than δ. Moreover, unlike $ERNIE$[11], which assigns an aligning sequence $\{e_1, ..., e_m\}$ to the token sequence $\{w_1, ..., w_n\}$, we select entities from tokens and choose parts of entities at random. In order to reflect the correlation between mathematical concepts and duplicated entities in mathematical questions, we mask these entities using [MASK] and predict the masked words using the corresponding hidden vectors.

In our training procedure, the strategy of token replacement is the same as that used in $BERT$. We replace the chosen entity by the following rules: (1) token [MASK] 80% of the time; (2) a random entity 10% of the time. Then, a transformer function is used to predict the original token or entity with cross-entropy loss.

Related Question Prediction In many natural language tasks such as question answering and natural language inference, understanding of the *relationships* of sentences is captured by next-sentence prediction in $BERT$[8]. Here, as the length of mathematical content tends to be shorter than that of other types of text corpus, especially as many questions have only one sentence, we design a strategy to generate the relationships of different mathematical questions with their mathematical concepts in knowledge graphs. We first extract mathematical entities e_i^q for question q. Then, we choose a pair of questions $< A, B >$ and evaluate their similarity using $sim(e_i^A, e_i^B)$, where sim is the similarity function for two sets. When the similarity value is larger then a threshold parameter γ, we treat B as a related question of A (i.e., $< A, B >$ is labeled as $Related$), otherwise $< A, B >$ is labeled as $NotRelated$ for training purposes. Despite the simplicity of the idea, we can choose various similarity functions to measure the relationship between two questions and demonstrate the benefit in our tagging problem.

Algorithm 1 Question relevancy rank

Require: a batch of pre-training samples $\mathcal{B}$, the number of candidate samples for ranking C
Ensure: $NDCG\_Loss$

```
P, Pair_Index= Random_Pairs_from_Batch(B,C)
Correlation_Scores = []
for pair in P do
    related_pair0 = FindNeighbors(pair[0])
    related_pair1 = FindNeighbors(pair[1])
    score = JaccardSimilarity(related_pair0, related_pair1)
    add score → Correlation_Scores
end for
Sequence_Output = BERT(B)
Candidate_Pairs = GetPairs(Sequence_Output, Pair_Index)
Candidate_Correlation_Scores = PearsonCorrelation(Candidate_Pairs)
NDCG_Loss = NDCG(Correlation_Scores, Candidate_Correlation_Scores)
```

Question Relevancy Rank

Mathematical concept questions and word problems are expressed differently for the same testing topic. Therefore, in our work, we integrate knowledge entities to train the encoding procedure to reconstruct token vectors. During the pre-training process, in each batch operation, we select n pairs of questions, $< q_i, q_j >$, and calculate their similarity using the sim function. The input collection for the transformer is $\{< q_1, q_2, v_{(1,2)} >, < q_1, q_3, v_{(1,3)} >, ..., < q_i, q_j, v_{(i,j)} >\}$, where $v_{(i,j)}$ is the similarity value, which is used as the score of each pair of questions.

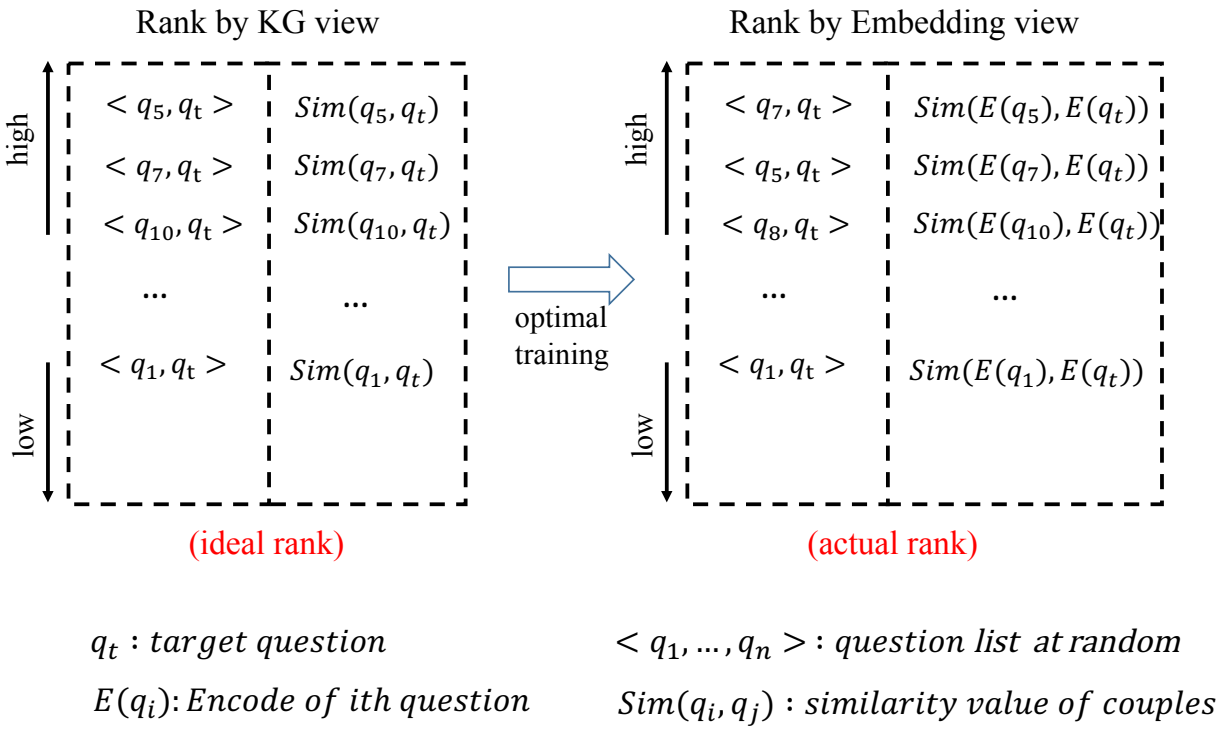


FIGURE 3 Relevancy rank task: our aim is to train word embedding to make questions much closer if they are similar based on the knowledge semantic measure

The goal of our transformer layers is to reconstruct the vectors of tokens and make two questions more similar after training, when they have close mathematical semantics. As shown in Figure 3, our encoders are:

$$\begin{aligned} \{\hat{w}_1^{(i)}, ..., \hat{w}_n^{(i)}\} &= MH - ATT(\{w_1^{(i-1)}, ..., w_n^{(i-1)}\}), \\ \{\hat{e}_1^{(i)}, ..., \hat{e}_n^{(i)}\} &= MH - ATT(\{e_1^{(i-1)}, ..., e_n^{(i-1)}\}). \end{aligned} \tag{1}$$

The question's semantic is encoded by its related entities; for simplicity, we use the function *mean* to calculate the vector space $\bar{e}_j^{(i)}$ of the question. In the fusion layer, we integrate the token sequence and semantic vector and compute the output embedding for each token and entity as follows:

$$\begin{aligned} \bar{e}_j^{(i)} &= mean(e_1^{(i)}, e_2^{(i)}, ..., e_k^{(i)}), \\ h_j &= \sigma(\tilde{W}_t^{(i)} \tilde{w}_j^{(i)} + \tilde{b}^{(i)}) + \alpha \cdot \bar{e}_j, \\ w_j^{(i)} &= \sigma(W_t^{(i)} h_j + b_t^{(i)}), \\ e_k^{(i)} &= \sigma(W_t^{(i)} h_j + b_e^{(i)}). \end{aligned} \tag{2}$$

In order to train the token and entity more closely in related questions, we use the ranking model $NDCG$ (normalized discounted cumulative gain) as the loss function. The process is shown in Algorithm 1. We define gain as the *Pearson correlation coefficient* of each pair of questions $\rho(q_i, q_j)$, use a collection $\mathcal{L}$ of $< q_i, q_j >$ ranked by coefficient value ρ, and set t as the index position of the collection.

Many existing similarity methods for use in knowledge graphs could be applied to $\rho(q_i, q_j)$, including string-based approaches [18], graph-based approaches [19], and embedding-based approaches [14,15,16]. Here, for simplicity, we use a graph-based method to compare two nodes based on their nearest neighbor. Given two questions q_i and q_j, E_i and E_j are sets of entities belonging to each question, respectively. For each entity e_i in set E, we obtain direct neighbor relationships $< e_n, r_{i,t}, e_i >$ and place these in an array called the *Jaccard* function according to the type of relationship.

$$\mathcal{D}(< q_i, q_j >, t) = \begin{cases} 2^{\rho(q_i,q_j)} - 1 & t = 0 \\ \dfrac{2^{\rho(q_i,q_j)} - 1}{log_2(t+1)} & t = 1 \end{cases} \tag{3}$$

$$\mathcal{DCG}(\mathcal{L}, f) = \sum_{t=1}^{n} y_{(t)}^{f} \mathcal{D}(r) \tag{4}$$

For simplicity, rank function f is defined as ranking collection $\mathcal{L}$ by $\rho(q_i, q_j)$ descent. Let the ideal DCG ($IDCG$) be defined by its original similarity value as follows:

$$\mathcal{IDCG}(\mathcal{L}, f') = \sum_{t=1}^{n} y_{(t)}^{f'} \mathcal{D}(r). \tag{5}$$

Here, function f' is not generated to the maximum DCG value of the collection but to rank collection $\mathcal{L}$ by the $\upsilon(q_i, q_j)$ value, which represents the original similarity of (q_i, q_j). Finally, the loss function is defined in every layer for training as $\mathcal{NDCG}(\mathcal{L}) = \frac{\mathcal{DCG}}{\mathcal{IDCG}}$.

4.4 | Topic Prediction Procedure

The topic prediction procedure of our model is similar to the fine-tuning process in $BERT$. We use a feature-based approach where fixed vectors of both tokens and entities are extracted from the pre-trained model. We use a special token [CLS] at the beginning of each question, and then we reuse the *mean* function to create semantic embedding with related entities. In contrast to the pre-training procedure, the topic prediction procedure focuses on the final target, mathematics question classification. In each integrator layer, we combine both token embedding information and entity embedding information.

Finally, we use the $sigmoid$ output function before the full connection layer and utilize binary cross-entropy as the loss function:

$$\ell_c(x, y) = L_c = \{l_{1,c}, ..., l_{N,c}\}^{\top}, \tag{6}$$

$$l_n = -\frac{1}{N} \sum_{i=1}^{N} [p_c y_{i,c} \cdot log\sigma(x_{i,c}) + (1 - y_{i,c}) \cdot log(1 - \sigma(x_{i,c}))], \tag{7}$$

where $c > 1$ for multi-label binary classification, and p_c is the weight of the positive value for the class c.

5 | EXPERIMENTS

In this section, we present experimental results obtained using our framework. First, we describe our dataset and experimental environment. Then, in comparison with several state-of-the-art methods, we evaluate our models with respect to precision and top-K. Finally, we demonstrate the effects of the parameters in various datasets.

5.1 | Pre-Training Dataset

Dataset: An overview of a real world Chinese mathematics dataset is given. We remove duplicated questions and simple questions with only a few terms. At last, we collect more than sixty thousand questions with 541 topic labels in Table 2.

For each label, we choose 80% of the data at random for training and use the remaining data for testing.

5.2 | Parameter Settings and Training Details

We use PyTorch and the 'bert-base-chinese' (Footnote) version of $BERT$ to implement the model. For pre-training, to accelerate the procedure, we set the maximum sequence length to 256 instead of 512, as the computation of self-attention is costly with respect to length. The two datasets are trained for 10 epochs on four GPUs (GTX 1080 Ti) with gradient accumulation per eight steps, which makes the batch size approximately 544. The random seed is 42, and the learning rate of Adam is 3e-5.

For fine-tuning, each model is trained for 60 epochs, saved as a checkpoint, and evaluated against the validation set every 10 epochs. The parameter of gradient accumulation steps in this procedure is 6, the batch size is 480, the learning rate of Adam is 5e-4, and the random seed is 2018. All other hyper-parameters are the same as in $BERT$.

TABLE 2 Summary statistics of *Chn-Math*

Chn-Math		Knowledge Graph	
Attribute	Value	Attribute	Value
# Questions	63913	# Entities	450
# Labels	541	# Edges	671
# Length	76.39± 43.56		
# Entities	5.21± 3.49		

We select the top three checkpoints based on their evaluation losses on the validation set, and report the averaged results on the test set.

5.3 | Experimental Evaluation

Based on the "all spot" and "special spot" scenarios, which are the same for all users, and the "favorite spot" scenario, which varies among users, we compare our method with a variety of competing methods grouped into three categories: classification methods, anomaly detection methods, and deep learning methods.

- $FastText$ learns vector representation and classifies texts. We train models in $D_{labeled}$ using the $FastText$ source code in github.
- $TextCNN$ uses multiple $kernels$ of different sizes to extract key information from sentences. We train models in $D_{labeled}$ using the $TextCNN$ source code in github.
- $BERT$ uses base model $BERT$ as the pre-trained model and refines the model during the fine-tuning procedure in the $Chn\text{-}Math$ dataset.
- $BERT_{MLM}$ implements the model under both $D_{unlabeled}$ and $D_{labeled}$ datasets. As $D_{unlabeled}$ has no label, we retrain the $BERT$ model only on the pre-training task.
- $KG\text{-}MTP_{EM}$ uses the same architecture as $BERT$ except the MLL, which is replaced by our entity mask strategy. In this model, we set only one mask task during the pre-training process.
- $KG\text{-}MTP_{EM,KG}$ is similar to $ERNIE$; we fuse the knowledge graph based on word and question granularity. In this model, we implement the semantic mask model in pre-training and in the fine-tuning procedure.
- $KG\text{-}MTP_{EM,RQP,RR}$ is based on $KG\text{-}MTP_{EM,KG}$ with two additional tasks: related question prediction and question relevancy rank. Datasets and all parameters are the same as in $KG\text{-}MTP_{EM}$ and $KG\text{-}MTP_{EM}$.

5.3.1 | Overall Performance

As shown in Table 3, $KG\text{-}MTP_{EM,RQP,RR}$ performs the best, significantly outperforming other baseline models. More in-depth analysis shows that the existing Chinese-only $BERT$ model performs a little worse than the mathematics pre-trained model $BERT_{MLM}$ in the top five, but is better in the top one and three. $KG\text{-}MTP$ models achieve significantly better results than $FastText$ and $TextCNN$. The main reasons are twofold. 1) the Entity encoding can capture both word-level and question-level representation. All the $KG\text{-}MTP$ methods are better than the original $BERT$-based models. 2) With the $KG\text{-}MTP$ methods, our proposed tasks such as related question prediction and relevant rank improve the performance effectively.

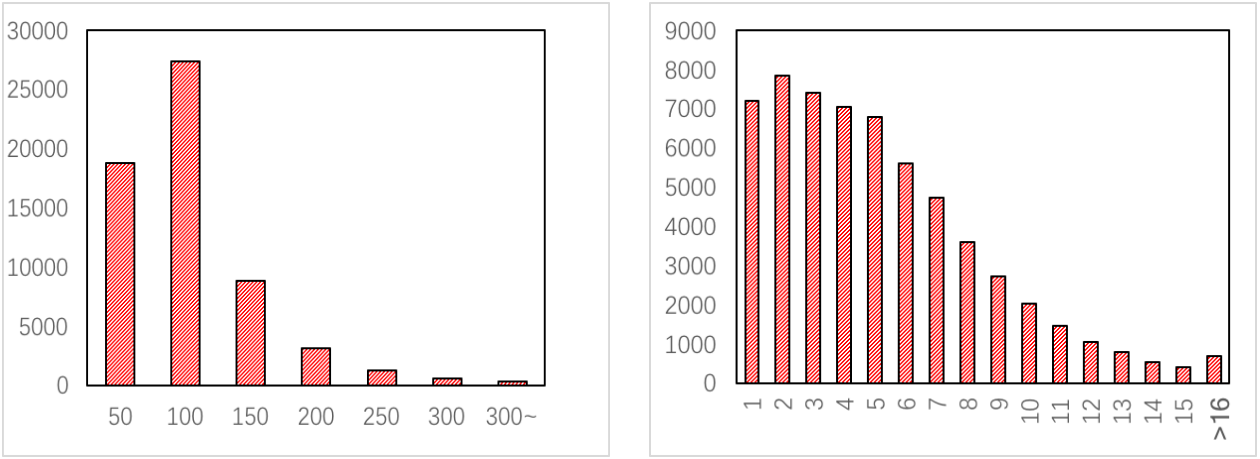
5.3.2 | Data Characteristic Sensitivity

In order to evaluate the effects of question content, we tested several main models (Table 3) on different split datasets. As shown in Figure 4, we evaluate the effectiveness of models with respect to two aspects: the length of questions and the number of

TABLE 3 Results of various models on D_{label}

Model	Precision		
	acc@1	acc@3	acc@5
$FastText$	38.44%	56.25%	62.80%
$TextCNN$	34.69%	55.34%	63.44%
$BERT$	55.57%	74.45 %	82.07%
$BERT_{MLM}$	56.10%	75.27 %	82.29%
$KG\text{-}MTP_{EM}$	56.19%	75.39%	82.93%
$KG\text{-}MTP_{EM,KG}$	56.87%	75.70 %	83.09%
$KG\text{-}MTP_{EM,RQP,RR}$	57.10%	75.53%	83.17%

entities in questions. For the length sensitivity experiment, we set the maximum length to 256 words (longer texts are set to 256), and divide $Chn\text{-}Math$ into six subgroups of equal length. As the number of items in different groups are different, and most questions contain fewer than 100 words, we select 1000 questions from each group at random. For entity sensitivity analysis, we also generate subgroups by one entity to five entities and consider additional groups as non-entities. Based on the results shown in Tables 4 and 5, we make the following observations. All the pre-trained models in $Chn\text{-}Math$ perform better than the un-pre-trained Chinese language model $BERT$ in all length groups. Although the $KG\text{-}MTP_{EM,RQP,RR}$ model with all tasks performs best on the whole dataset, it still does not perform quite as well as other simpler models. For instance, $KG\text{-}MTP_{EM}$ performs the best in Top-1 in G2, and $BERT_{MLM}$ has the highest accuracy in Top-5 in G4, which demonstrates that the performances of various pre-trained models are not stable.



(a) Distribution of question lengths.

(b) Count of entities in questions.

FIGURE 4 Question features distribution of dataset.

We further compare the prediction accuracy of our best-performing model $KG\text{-}MTP_{EM,RQP,RR}$ ($KG\text{-}MTP$ for short) with those of the three baseline methods, i.e., $FastText$, $TextCNN$, and $BERT$ ($BERT$ for short) , by keeping the parameters the same. As shown in Table 4, all the models performed worse as the length of questions increased, indicating that an increasing number of terms increased the amount of noise information, especially in G4, which might contain word problem questions with many descriptive words. In terms of entity size, as shown in Table 5, models perform better when questions contain more entities. This is probably because entity size is important in questions. More entities could facilitate richer mathematical semantics.

TABLE 4 Performance evaluation for various question lengths

Method	G1:$length \leq 64$			G2:$65 < length \leq 128$			G3 :$129 < length \leq 192$			G4 :$193 < length \leq 256$		
	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5
$FastText$	42.35	55.46	66.37	35.81	52.34	68.13	29.99	47.29	64.34	29.57	41.07	60.72
$TextCNN$	41.17	56.25	65.40	32.33	51.38	62.55	28.44	44.87	61.32	23.91	38.71	59.75
$BERT$	52.40	73.88	82.59	58.30	76.35	83.15	55.00	73.66	81.78	49.40	70.38	78.86
$BERT_{MLM}$	54.50	74.04	82.34	59.19	77.27	83.78	55.50	75.58	82.21	51.60	71.82	80.60
$KG\text{-}MTP_{EM}$	55.10	74.53	82.87	**59.80**	**78.07**	84.32	55.20	**75.78**	81.96	51.10	71.88	**81.07**
$KG\text{-}MTP_{EM,KG}$	56.30	74.07	83.24	59.30	78.04	**84.78**	**57.09**	74.76	**83.34**	**52.00**	**72.76**	80.90
$KG\text{-}MTP_{EM,RQP,RR}$	**57.90**	**75.55**	**83.88**	57.69	75.45	83.33	55.88	75.59	82.70	51.70	71.44	80.72

TABLE 5 Performance evaluations with various entities

Method	non			1			2			3			4		
	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5	acc@1	acc@3	acc@5
$FastText$	43.23	50.6	62.31	47.34	55.34	67.39	47.25	56.22	70.85	48.56	56.74	71.29	47.24	58.7	71.07
$TextCNN$	42.56	51.04	60.55	45.88	56.21	65.19	48.96	57.84	66.61	44.35	57.21	68.93	46.58	56.87	70.50
$BERT$	49.70	69.12	78.35	**58.69**	75.13	82.34	57.49	73.95	82.24	57.09	74.21	81.60	55.10	73.76	83.05
$BERT_{MLM}$	49.81	69.10	**78.79**	58.29	76.17	82.31	58.19	75.40	82.87	58.60	74.42	82.12	57.25	77.35	84.25
$KG\text{-}MTP_{EM}$	**50.78**	68.24	78.13	57.99	76.15	**83.73**	56.89	76.41	82.51	58.09	73.67	80.79	58.69	77.33	**84.70**
$KG\text{-}MTP_{EM,KG}$	48.13	68.90	78.35	58.59	**77.35**	83.45	**59.19**	**76.86**	**84.42**	**60.19**	74.48	82.86	59.59	77.61	84.24
$KG\text{-}MTP_{EM,RQP,RR}$	50.54	**69.23**	78.68	58.19	74.67	82.71	58.80	74.95	83.69	59.09	**74.93**	**83.22**	**59.69**	**78.07**	84.15

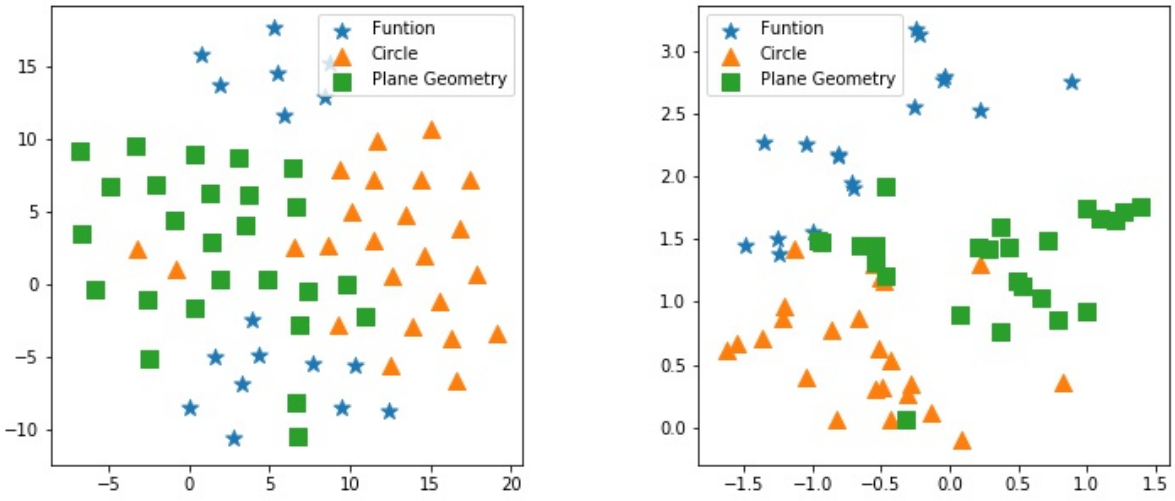
TABLE 6 Keywords of different labels (in part)

Labels	Keywords
Function	Variable, Constant, Inverse Function, Quadratic Function, Dependent Variable, Trigonometric Function, ...
Circle	Arc, Radius, Diameter, Curve Circumscribed Circle, Sector, External Cutting, ...
Plane Geometry	Quadrilateral, Straight Line, Square, Congruent Triangle, Right-Angle Trapezoid, Diamond, ...

5.3.3 | Term and Entity Visualization

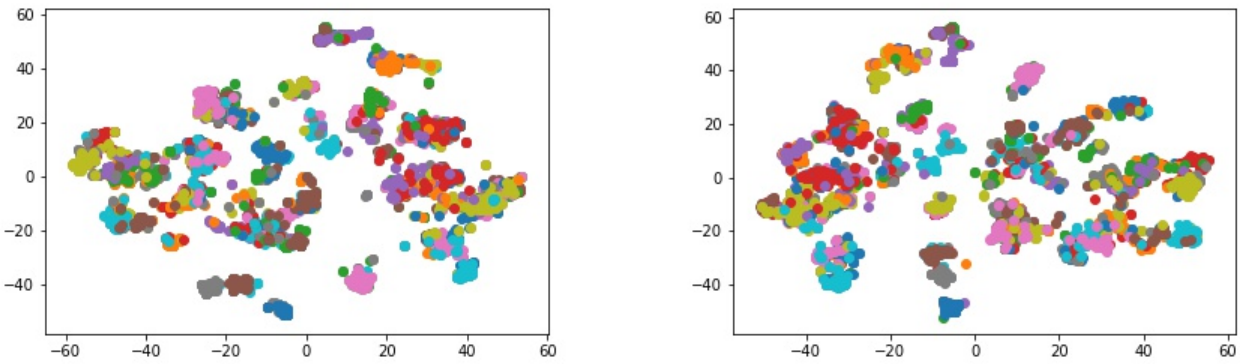
In this section, we demonstrate the effectiveness of the knowledge-based encoding method in our pre-trained model through word-level and question-level visualization. As shown in Figure 5 and 6,we use the t-SNE tool to visualize word-level and question-level visualization by $BERT$ and $KM\text{-}MTP$[20]. We choose several labels, some of which are close in terms of mathematical semantics, whereas others are relatively different. We choose keywords for each class via the TF-IDF method, and discard stop words and LaTeX symbols (which have been translated into terms). Some of the selected words are shown in Table 6.

To provide an illustrative visualization of the question vectors learned by $BERT_{chn}$ and $KG\text{-}MTP_{EM,RQP,RR}$, we select 350 distinct labels and their related questions and obtain question vectors from the output layers. The results show that $KG\text{-}MTP_{EM,RQP,RR}$ can learn more discriminate question vectors, which means that most questions are closer to each other if they belong to the same label. Question vectors integrating knowledge information appear more similar than original vectors.



(a) $BERT_{chn}$ (b) $KG\text{-}MTP_{EM,RQP,RR}$

FIGURE 5 The t-SNE visualization of word vectors



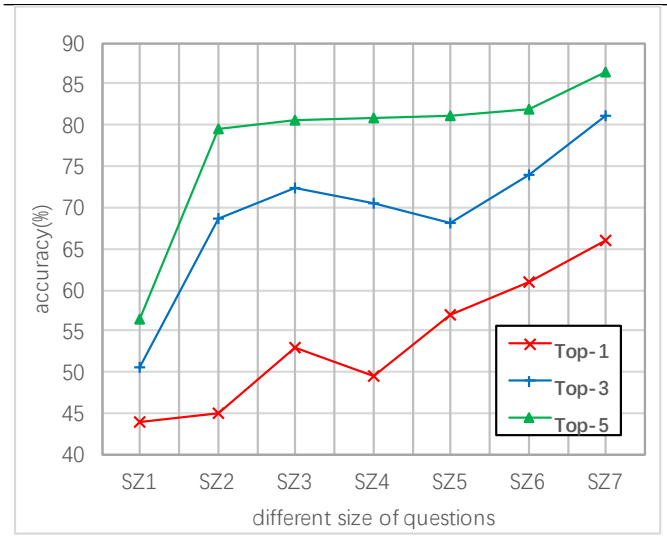
(a) $BERT_{chn}$ (b) $KG\text{-}MTP_{EM,RQP,RR}$

FIGURE 6 The t-SNE visualization of question vectors (perplexity=30, init='random')

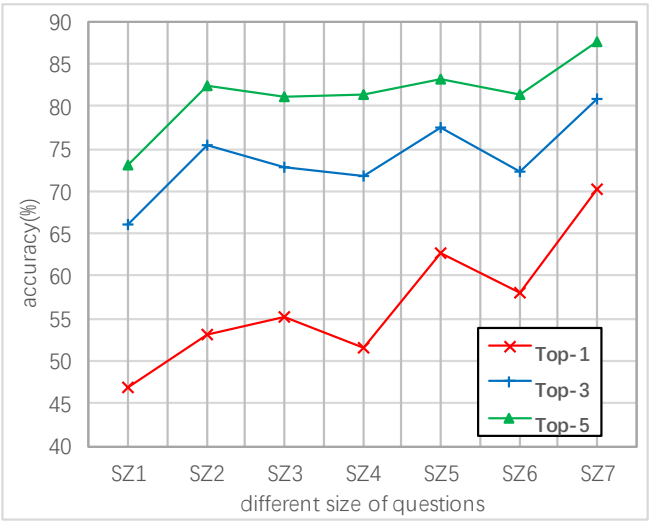
5.3.4 | Data Size Sensitivity

In order to evaluate the effect of the size of the labeled data, we tested the performance of the original $BERT_{chn}$ and $KG\text{-}MTP_{EM,RQP,RR}$ with different sizes of training data. In dataset $Chn\text{-}Math$, each label has various numbers of question instances, from 30 to 231. Therefore, we generate several datasets according to the number of questions and evaluate the prediction accuracy. The results are shown in Figure 7.

We evaluate the prediction accuracy in various groups to demonstrate effectiveness by data size. Here, we split the data into seven groups (named SZ) based on the number of questions, e.g., $SZ1$ = [30,60], $SZ2$ = [60,90], and $SZ3$ = [90,120]. The results illustrate that prediction performance is improved given more instances in all Top-1, Top-3, and Top-5 targets. We note that our model $KG\text{-}MTP$ performs better in the $SZ1$ group, which indicates the effectiveness of knowledge entities in a situation with few instances. Similarly, compared with the un-pre-trained $BERT$ model, $KG\text{-}MTP$ performs relatively well.



(a) Evaluation by $BERT$.



(b) evaluation by $KG\text{-}MTP$.

FIGURE 7 Prediction accuracy for various data sizes.

5.4 | Insights and Results Discussion

To better understand the performance of various models, we summarize our experimental results and also discuss performance in different situations.

(1) **Difficulty of mathematics understanding**. Mathematical questions are usually shorter than normal text and contain language, symbols, and pictures that are heterogeneous in text. The goals of mathematics understanding are counting, sorting, matching, seeking patterns, making connections, and recognizing relationships. Therefore, external knowledge is necessary to extract the latent semantics of questions and has been utilized in some short text classification models. In this work, for simplicity, we ignore figures and choose questions without figures, although obviously figures may contain some mathematical information.

(2) **Effectiveness of entities**. Our experiments showed that knowledge graph entities could improve the accuracy of prediction, which suggests that the information fusion method could result in richer features of questions. However, to the best of our knowledge, there is no public mathematics knowledge graph specifically in Chinese. It is a difficult task to extract elements from mathematics textbooks and generate their relationships. Moreover, according to our experience, mathematical symbols written in LaTeX format are also significant for prediction.

(3) **Effectiveness of question length**. There are many different styles of question that can be used very effectively to develop knowledge of mathematics and mathematical skills, including algebraic expressions, linear equations, fractions, and functions; thus, some questions are simple and short whereas some others are complex and longer. Based on our evaluation, simple questions are relatively easy to classify.

(4) **Effectiveness of data size**. More instances for training would improve the performance of pre-trained models such as $BERT$ and $KG\text{-}MTP$. It is obvious that a larger data corpus would also improve performance. Pre-trained models with knowledge information fusion perform better, especially in situations with fewer instances, probably because knowledge has a more important role when the data are insufficient. For common mathematical questions, there are enough examples for training, and so the role of knowledge entities is not very important. However, when the size of the dataset is relatively small, as is the case for some unusual mathematical concepts, external information is more important for improving the accuracy of models.

6 | CONCLUSION AND FUTURE WORK

In this work, we have investigated the problem of question topic prediction. We proposed a framework based on a pre-trained model, integrating external information in a mathematical knowledge graph. To improve the expressive of model, three pre-training tasks is designed. And we designed a novel topic prediction task for mathematical questions in Chinese. We also built

a real dataset with human-defined labels and set up a knowledge graph for mathematics concepts used in middle school education. Extensive experiments show that our $KG\text{-}MTP$ model significantly outperforms all baselines, including $BERT$ and its extended models.

There are several directions for future work. First, we only used textual information to describe semantics of questions. Other information such as figures will be considered in further research. Second, our current work did not consider the structure of mathematical formulas in LaTeX, which usually contains latent concepts. Third, we plan to add more data besides current questions in middle school education and update our mathematical knowledge graph accordingly. We will make the dataset available to the public and hope that more researchers will make use of it. Furthermore, although our experiments were on Chinese data, our method is generalizable and we plan to apply it to questions in English and to other subjects such as physics and chemistry.

CONFLICT OF INTEREST

All the authors declare that they have no conflict of interest.

References

1. Rittle-Johnson B, Alibali MW. Conceptual and procedural knowledge of mathematics: Does one lead to the other?. *Journal of educational psychology* 1999; 91(1): 175.

2. Song G, Ye Y, Du X, Huang X, Bie S. Short text classification: A survey. *J Multimed* 2014; 9(5): 635.

3. Chen M, Jin X, Shen D. Short text classification improved by learning multi-granularity topics. In: ; 2011.

4. Tang J, Qu M, Mei Q. Pte: Predictive text embedding through large-scale heterogeneous text networks. In: ACM. ; 2015: 1165–1174.

5. Wang G, Li C, Wang W, et al. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174* 2018.

6. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* 2016.

7. Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. In: . 33. ; 2019: 7370–7377.

8. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* 2018.

9. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237* 2019.

10. Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *arXiv preprint arXiv:1907.10529* 2019.

11. Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. ERNIE: Enhanced Language Representation with Informative Entities. *arXiv preprint arXiv:1905.07129* 2019.

12. Sun Y, Wang S, Li Y, et al. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv preprint arXiv:1904.09223* 2019.

13. Wang Q, Mao Z, Wang B, Guo L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans Knowl Data Eng* 2017; 29(12): 2724–2743.

14. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: ; 2013: 2787–2795.

15. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: ; 2014.

16. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: ; 2015.

17. He S, Liu K, Ji G, Zhao J. Learning to represent knowledge graphs with Gaussian embedding. In: ACM. ; 2015: 623–632.

18. Ngomo ACN, Auer S. LIMES–a time-efficient approach for large-scale link discovery on the web of data. In: ; 2011.

19. Raimond Y, Sutton C, Sandler MB. Automatic Interlinking of Music Datasets on the Semantic Web.. *LDOW* 2008; 369.

20. Maaten Lvd, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008; 9: 2579–2605.

21. Sriram B, Fuhry D, Demir E, Ferhatosmanoglu H, Demirbas M. Short text classification in Twitter to improve information filtering. In: ACM. ; 2010: 841–842.