Boise State University
ScholarWorks

Educational Technology Faculty Publications and Presentations | Department of Educational Technology

7-2023

S-KMN: Integrating Semantic Features Learning and Knowledge Mapping Network for Automatic Quiz Question Annotation

Jing Wang
Central China Normal University

Hao Li
Central China Normal University

Xu Du
Central China Normal University

Jui-Long Hung
Boise State University, andyhung@boisestate.edu

Shuoqiu Yang
Central China Normal University

Follow this and additional works at: https://scholarworks.boisestate.edu/edtech_facpubs

Part of the Educational Technology Commons, and the Instructional Media Design Commons

Publication Information

Wang, Jing; Li, Hao; Du, Xu; Hung, Jui-Long; and Yang, Shuoqiu. (2023). "S-KMN: Integrating Semantic Features Learning and Knowledge Mapping Network for Automatic Quiz Question Annotation". *Journal of King Saud University - Computer and Information Sciences, 35*(7), 101594. https://doi.org/10.1016/j.jksuci.2023.101594

Journal of King Saud University – Computer and Information Sciences 35 (2023) 101594

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

S-KMN: Integrating semantic features learning and knowledge mapping network for automatic quiz question annotation

Jing Wang [a], Hao Li [a,*], Xu Du [a], Jui-Long Hung [b,c], Shuoqiu Yang [a]

[a] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China
[b] National Engineering Laboratory For Educationgal Big Data, Central China Normal University, Wuhan 430079, China
[c] Department of Educational Technology, Boise State University, Boise, ID 83725, USA

ARTICLE INFO

Article history:
Received 23 March 2023
Revised 17 May 2023
Accepted 17 May 2023
Available online 8 June 2023

Keywords:
Automatic quiz question annotation
Knowledge mapping network
Knowledge attribute graph
Latent knowledge space
Semantic-knowledge features

ABSTRACT

Quiz question annotation aims to assign the most relevant knowledge point to a question, which is a key technology to support intelligent education applications. However, the existing methods only extract the explicit semantic information that reveals the literal meaning of a question, and ignore the implicit knowledge information that highlights the knowledge intention. To this end, an innovative dual-channel model, the Semantic-Knowledge Mapping Network (S-KMN) is proposed to enrich the question representation from two perspectives, semantic and knowledge, simultaneously. It integrates semantic features learning and knowledge mapping network (KMN) to extract explicit semantic features and implicit knowledge features of questions,respectively. Designing KMN to extract implicit knowledge features is the focus of this study. First, the context-aware and sequence information of knowledge attribute words in the question text is integrated into the knowledge attribute graph to form the knowledge representation of each question. Second, learning a projection matrix, which maps the knowledge representation to the latent knowledge space based on the scene base vectors, and the weighted summations of these base vectors serve as knowledge features. To enrich the question representation, an attention mechanism is introduced to fuse explicit semantic features and implicit knowledge features, which realizes further cognitive processing on the basis of understanding semantics. The experimental results on 19,410 real-world physics quiz questions in 30 knowledge points demonstrate that the S-KMN outperforms the state-of-the-art text classification-based question annotation method. Comprehensive analysis and ablation studies validate the superiority of our model in selecting knowledge-specific features.
© 2023 The Author(s). Published by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Quiz questions, as the most typical learning resources for assessing students' mastery of knowledge and diagnosing students' learning blind spots (Ahmed et al., 2022; Wu et al., 2020b), are widely applied in all levels of educational scenarios to achieve personalized learning (Zhou et al., 2018; Abdous et al., 2012). The realization of some intelligent education applications benefits from question sets labeled with knowledge points, such as knowledge tracing (Liu et al., 2022), educational resource recommendation (Wang et al., 2022) and knowledge diagnosis (Zhang et al., 2020a). These question sets with knowledge labels are formed by assigning appropriate knowledge points to the questions. This process is called the question annotation task (Sun et al., 2018). Traditionally, question annotation is usually completed manually by experienced teachers or domain experts (Wu et al., 2020b; Silva et al., 2018), this process is time-consuming and expensive. With the rapid development of deep learning approaches, most researchers have designed novel automatic question annotation models to improve annotation efficiency and accuracy (Almuzaini and Azmi, 2022; Kurdi et al., 2020).

Automatic question annotation is usually regarded as a specific task of text classification, and knowledge points, as knowledge labels, are automatically assigned to questions (Yilmaz et al., 2019). Hence, natural language processing (NLP) algorithms and deep learning (DL) techniques are widely used in this task, mainly

* Corresponding author.
E-mail addresses: wangjing_dl@mails.ccnu.edu.cn (J. Wang), lihao205@ccnu.edu.cn (H. Li), duxu@mail.ccnu.edu.cn (X. Du), andyhung@boisestate.edu (J.-L. Hung), yshuoq@mails.ccnu.edu.cn (S. Yang).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101594
1319-1578/© 2023 The Author(s). Published by Elsevier B.V. on behalf of King Saud University.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

focusing on two aspects: 1) The first is to learn more powerful semantic representations of questions through neural language models, such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), ELMo (Jia et al., 2019), BERT (Devlin et al., 2019), and XLNet (Zhang et al., 2020c). These methods were commonly adopted to extract explicit semantic representations of quiz questions. In addition, some researchers incorporate domain information or introduce external knowledge bases to extract additional features (Paramasivam and Nirmala, 2022), such as syntax features (Wang et al., 2017), lexical features (Liu et al., 2018), and keyword-relevant features (Sun et al., 2018). 2) The second is to design effective annotation algorithms based on neural network models such as CNNs (Liao et al., 2017; Wang et al., 2017), RNNs (Kadhim, 2019; Qin et al., 2020; Jang et al., 2020), GNNs (Wu et al., 2020c; Li et al., 2020c), and ensemble models (Ahmad et al., 2022; Li et al., 2020a). CNNs can learn local features from temporal or spatial data, RNNs can learn sequential correlations, the attention mechanism can highlight important information in multi-dimensional feature representations by setting adaptive weights, and GNNs can model the correlation between words and questions. A combination of multiple neural networks can take full advantage of their respective advantages to improve the accuracy of question annotation.

Although the existing research has achieved encouraging annotation results by extracting explicit semantic information of the question text, it is difficult to annotate accurate knowledge points for some questions with similar semantics but different knowledge intention. As shown in Fig. 1, the surface semantics of q_1 and q_2 are "the objects move in a straight line at a constant speed", but one question is concerned with the speed of the object, so the question is labeled as "Speed". Another question is to examine the force acting on the object, so the question belongs to the "Force". It can be concluded that the explicit semantic features of the question text cannot fully characterize the question, which may lead to misjudgment of knowledge points. Hence, in addition to considering the semantic information, further identifying the knowledge intention of the question is the key to improving the accuracy of question annotation; this is also not considered in the existing research.

To grasp the knowledge intention of quiz questions, we simulate the process of an annotator thinking about a question, as shown in Fig. 1. First, after reading through the entire question text, the annotator focuses on some key terms in the question (Hassani et al., 2022), such as "speed", "constant speed" and "m/s" in q_1 and q_2. Compared with other terms, these keywords provide more useful information and are widely applied in multiple knowledge scenarios; these keywords are referred to as "knowledge attribute words" in this work. However, words contain limited information, the annotator may continue to analyze the contextual information of these knowledge attribute words to clarify their knowledge scenarios. For instance, the context of "constant speed" in q_2 is a "smooth horizontal surface", which is different from "in a straight line" in q_1. Different contextual information endows "constant speed" with different knowledge meanings, which is helpful for analyzing the knowledge intention of questions. The annotator understands the knowledge information expressed by the question by relying on the rich contextual information of knowledge attribute words. Then, given the known knowledge representation and the existing knowledge structure, the annotator selects the most appropriate knowledge point for a question after knowledge information processing. Therefore, the knowledge connotation in the question is different from the explicit semantic information. It is acquired through certain cognitive processing on the basis of understanding the basic semantics.

Inspired by the thought process of an annotator labeling questions, this paper aims to design a novel model to understand the knowledge intention of questions to enrich the question representation from two dimensions of knowledge and semantics. This paper is divided into the following three main research objectives: (1) The first is to design a knowledge representation method similar to the question understanding of the annotator. The knowledge attribute words and contextual information in the question are combined to interpret the knowledge connotation of questions. (2) The second objective is to construct a deep learning-based model that extracts knowledge features of questions according to the subject knowledge space; that is similar to the annotator cognitive processing. (3) The third objective is to develop a feature fusion method that comprehensively considers explicit semantic information and implicit knowledge information to achieve more accurate question annotation.

To achieve the above purposes, we propose a new dual-channel network framework, the Semantic-Knowledge Mapping Network (S-KMN), which integrates semantic features learning and knowledge mapping network (KMN) to extract semantic features and knowledge features in the question text. The overall framework of the KMN for extracting knowledge features is shown in Fig. 2, which simulates the thought process of an annotator labeling a question. First, a knowledge attribute vocabulary is constructed to select words with knowledge connotations from a large number of learning resources. On this basis, each question is represented as a heterogeneous knowledge attribute graph to visualize knowledge

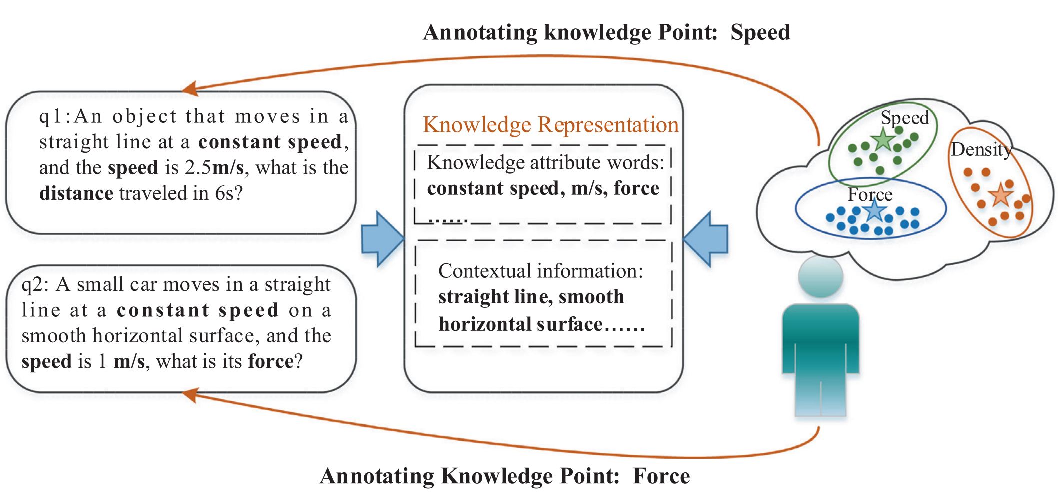

Fig. 1. The thought process of an annotator assigning the knowledge point to a question.

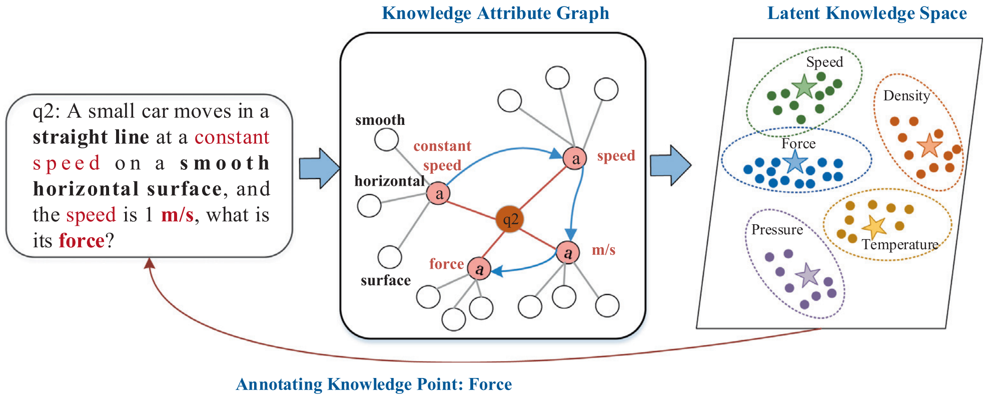

Fig. 2. An illustrative architecture of the KMN that simulates the thought process of an annotator annotating a quiz question.

information (Zeng et al., 2021). The graph makes full use of the context and sequence information of knowledge attribute words to generate the knowledge representation of questions. Then, a latent knowledge space is constructed based on the scene association matrix to analyze the application scenarios of knowledge points. The knowledge representation of each question is mapped to the latent knowledge space through a learnable projection matrix to extract knowledge features. To integrate semantic features and knowledge features, an attention mechanism is utilized to assign the adaptive learning weight for each dimensional feature, thus forming the comprehensive question representation. Extensive experiments show that the S-KMN models outperform the state-of-the-art text classification-based question annotation method on real-world physics quiz questions.

The main contributions of this paper can be summarized as follows:

- Semantic-Knowledge Mapping Network, which simulates the thought process of an annotator labeling a question, is proposed to integrate the semantic and knowledge features in questions. This framework enriches the feature representation of questions from both semantic and knowledge perspectives and improves the annotation performance.
- A knowledge mapping network is designed to extract implicit knowledge features and understand the knowledge intention of questions, similar to the cognitive processing of the annotators. This network realizes the extraction of knowledge-specific features.
- A latent knowledge space is constructed with the scene association matrix as a base to represent the connotation of subject knowledge. The closer the knowledge features is in the latent knowledge space, the more similar the knowledge connotation of questions.
- Comparative experiments on real-world physics questions are conducted. Compared to the state-of-the-art text classification-based question annotation method, our model achieves significant improvements for quiz question annotation tasks by incorporating implicit knowledge features.

The remainder of this paper is organized as follows. Section 2 discusses the related work regarding semantic feature representation and question annotation. Section 3 defines the related concepts proposed in this paper. The overall framework of the S-KMN is described in Section 4, which includes descriptions of the context-aware knowledge representation, knowledge features extraction and semantic-knowledge features fusion. To verify the effectiveness of S-KMN in extracting implicit knowledge features, extensive experiments are performed in Section 5. Conclusions and suggestions for future work are discussed in Section 6.

2. Related work

Automatic question annotation is the task of mapping questions to a label system and effectively associates pre-determined knowledge points to questions, following a certain criterion. Since quiz questions are in the form of text, traditional text classification methods are applied to automatic question annotation tasks. The feature representation of question text is enriched by integrating domain information. Among them, an efficient feature representation and a annotation algorithm are key steps to achieve domain adaption (Liu and Guo, 2019; Paramasivam and Nirmala, 2022). Hence, the related work discussed in this section mainly focus on the following two main subsections.

2.1. Feature representation of question text

General text feature engineering starts from the Bag of words (BOW), which emphasizes the frequency information of terms (Wang and Manning, 2012). N-grams is a promotion of BOW (1-gram). Although the co-occurrence frequency of words is taken into consideration, it disregards the contextual information between words and yields to high sparsity. To address these challenges, the distributed representations are proposed to map each word or phrase to a latent semantic space. Each word in the vocabulary is represented by a V-dimension vector, which can capture richer semantic from context and facilitate text understanding. The mainstream methods includes Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and FastText (Joulin et al., 2017). With the effectiveness of the pre-trained language model, the context-dependent representation models such as ELMo (Ethayarajh, 2019), GPT (Naseem et al., 2021), BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019) are proposed to learn text representations. In addition to considering implicit text representation, most of the existing work combines both explicit and implicit representations to obtain rich textual features, using part-of-speech (POS) tagging and syntactic roles annotation to extract lexical or syntactical features. For instance, Wang et al. (2017) combined explicit and implicit representations of short text through convolutional neural networks. Zhang et al. (2020c) proposed semantics-aware BERT, which incorporates the pre-trained semantic role annotation over a BERT backbone.

Considering the unique characteristics of specific domains, scholars have proposed customized approaches to enrich feature representations by adding extra features (Khot et al., 2020; Xu et al., 2019). This method is usually applied by adding domain information or utilizing an external knowledge base. For instance, Wang et al. (2017) introduced an existing knowledge base to capture more semantic and syntax information. Liu et al. (2018) applied the semantic similarity between words to extract semantic features, and used sequential pattern-mining algorithm to obtain lexical features for English question classification. Sun et al. (2018) proposed a position-based attention model and keywords-based model to consider the integration of the query, option, and answer for the representation of multiple-choice questions. Qiao and Hu (2019) applied a comprehensive set of features including lexical, syntactic and semantic features in the cognitive domains. Yang et al. (2018) proposed a feature-enhanced attention network to leverage the unigram features, the part of speech features, and the word position features in sentiment classification. Mohasseb et al. (2018) exploited the question structure to present a grammar-based approach for questions categorization. Therefore, the latest research which incorporates the unique information of the respective domains into textual feature representation, is verified to improve the classification accuracy, such as clinical text classification relying on biomedical knowledge sources (Yao et al., 2019a), sentiment analysis focusing on opinion word and specific aspect expression (Zhao et al., 2021; Yang et al., 2014), legal text classification using domain concepts (Chen et al., 2022), and question classification highlighting the importance of the answer to the question (Sun et al., 2018).

Inspired by the integration of domain information, this work aims to enrich the feature representation of questions by extracting domain-specific features related to knowledge points. Although some studies have similar practices, they need to consider information outside the question, such as answers and question types (Sun et al., 2018), and ignore the knowledge information of the question itself. To extract the knowledge-specific features, we pay attention to the question itself, and comprehensively consider the knowledge attribute words and their contextual information in the question.

2.2. Automatic question annotation

Automatic Question annotation is considered as a specific task in text classification (Lv et al., 2020). Traditional machine learning methods have been utilized to design different forms of classifiers to improve the accuracy of question annotation. With the substantial success of deep learning algorithms, some deep learning models are applied to generate effective classifiers, such as sequence representation models (Liao et al., 2017; Liu and Guo, 2019; Wu et al., 2020a), structural representation models (Ye et al., 2017; Huang et al., 2021), and attention-based models (Yang et al., 2016; Meškelė and Frasincar, 2020). They have been applied in question annotation tasks and achieved the state-of-the-art performance. Convolutional neural networks (CNNs) have been shown to achieve impressive results in the field of natural language processing. Kalchbrenner et al. (2014) proposed the Dynamic Convolutional Neural Network (DCNN) for question classification and sentiment prediction. Guo et al. (2019) made a simple modification for a simple CNN with the little tuning of the hyperparameters for various classification tasks (TextCNN). Although CNNs show good performance in the question annotation task, the sequential correlations and the position information of the words in a question may be ignored. Due to the ability to process sequences of arbitrary length, Recurrent neural networks (RNNs) are used more frequently in question annotation. For instance, Wang and Nyberg (2015) used a stacked Bidirectional Long-Short Term Memory (BiLSTM) network to calculate the relevance scores between the words from question and answer sentences. Yin et al. (2016) presented an attention-based convolutional neural network for sentence pair modeling tasks (ABCNN). Li et al. (2020a) implemented an architecture similar to ABCNN to represent the question text. Qin et al. (2020) considered the influence of neighborhood characteristics on text tagging, and combined BiGRU and self-attention mechanism to aggregate k-nearest-neighbor documents into feature vectors for text tagging. Zeng et al. (2021) utilized a bi-directional GRU (BiGRU) to encode the paragraph representation and the answer respectively. To highlight the important information in textual features, the attention mechanism is applied to assign important weights to keywords in questions.

Because different neural network models have their own advantages and disadvantages, some researchers have tried to combine the CNNs, RNNs and attention mechanism, which make full use of their respective advantages to achieve better performance in the question tagging task. Liu et al. (2019) combined a convolutional neural network, attention mechanism, and recurrent neural network to propose a novel deep neural network model named Attention-Based BiGRU-CNN network (ABBC) for Chinese question classification. Liu and Guo (2019) proposed a novel architecture that combines BiLSTM, the attention mechanism and the convolutional layer (AC-BiLSTM) for question classification. Compared with the single structural models, the combination of multiple models or mechanisms can make full use of the advantages of their respective models to achieve better performance in the question annotation task.

Graph neural networks (GNNs) (Wu et al., 2020c; Zhou et al., 2020) as powerful tools are applied to learn text embedding through aggregating neighborhood features in a graph structure and they have shown state-of-the-art performance compared with many popular neural network models (Bastings et al., 2017). Many variants of GCNs and GNNs have been proposed and explored on a variety of tasks, such as graph classification (Lee et al., 2019; Ying et al., 2018), link prediction (Kipf and Welling, 2017; Zhang and Chen, 2018) and node classification (Li et al., 2020c). For instance, Zhang et al. (2018) proposed a novel end-to-end learning framework over the knowledge graphs to exploit the structure information realizing logic reasoning. To model knowledge evolution of students in interactive online question pools, Li et al. (2020b) presented a new convolutional GNN model to achieve better student performance prediction by constructing the student-interaction-question network.

Although many deep learning models have been well studied in the question annotation task, the knowledge point is only used as a classification category, ignoring its knowledge characteristics. Therefore, we present a novel network structure to learn the knowledge information hidden in the question by focusing on the words with knowledge characteristics and the application context in the question. Distinguished from previous work, which induces an external knowledge base or additional text information, we project the knowledge representation of questions to the latent knowledge space to extract knowledge-relevant features, which only depend on the question text itself.

3. Preliminaries

One of the main contributions is to define the related concepts for question annotation task in this work. These concepts include knowledge attribute vocabulary, knowledge attribute graph, scene association matrix, latent knowledge space, and knowledge features. In this section, we define these concepts in detail, and relevant notations are listed in Table 1.

Definition 1. Knowledge Attribute Vocabulary. Some words with clearer knowledge characteristics, such as "Speed", "Constant Speed" and "m/s", are often applied in knowledge scenarios to

Table 1
Notations and explanations.

Notations	Explanations
$\mathscr{Q}$	Questions
$\mathscr{G}$	Knowledge attribute graph of each question
k_i	The i-th knowledge point
$\mathscr{V}$	Nodes
s_i	The i-th knowledge scenario
$r_{a_i}^{s_j}$	The delay factor
t_i	The i-th word
$\mathscr{E}$	Edges
a_i	The i-th knowledge attribute word
$\mathscr{O}$	Order information of knowledge attribute words
w_{ij}	Association weight
C_{win}	The sliding window
$\mathbf{A}$	Scene association matrix
$\mathbf{Q}_k^*$	Knowledge features
$\mathbf{S}$	Latent knowledge space
$\mathbf{Q}_s^*$	Semantic features

explain knowledge points. These words are called knowledge attribute words in this paper, and they are denoted as $\mathscr{K}^a = \{a_1, a_2, \cdots, a_m\}$, where m denotes the number of words in the knowledge attribute vocabulary; these words are selected from many learning resources, including textbooks, supplementary books, and quiz questions, etc.

Definition 2. Knowledge Attribute Graph. A heterogeneous knowledge attribute graph for each question is constructed, denoted as $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{O})$. It captures the relationship among a question, knowledge attribute words, and their neighbor words to highlight the knowledge information. As presented in Fig. 2, a question is the center of the graph, the knowledge attribute words of questions are directly connected nodes, and the neighborhood words are connected with the corresponding knowledge attribute words. $\mathscr{O}$, which denotes the order relationship of these words in a question, is incorporated to form the context-aware knowledge representation.

Definition 3. Scene Association Matrix. Some knowledge attribute words are applied in multiple knowledge scenarios, and the importance of these words may vary across different knowledge scenarios. Denoted as w_{ij}, the association weight is calculated to represent the interrelation of words a_i and knowledge scenarios s_j. The scene association matrix is composed of the association weights between all knowledge attribute words and knowledge scenarios; this matrix is expressed as $\mathbf{A} = \{[w_{11}, w_{12}, \cdots, w_{1k}], \cdots, [w_{m1}, w_{m2}, \cdots, w_{mk}]\}$. k denotes the number of knowledge scenarios, and this number equals the number of knowledge points.

Definition 4. Latent Knowledge Space. The latent knowledge space $\mathbf{S}$ using the scene association matrix as a base is constructed to enrich the connotation and extension of knowledge points; this space is expressed as $\mathbf{S} = \{s_1, s_2, \cdots, s_k\}, s_i$ is the i-th scene base. In this space, knowledge characteristics between knowledge vectors with closer distances are more similar.

Definition 5. Knowledge Features. Knowledge features, as a new dimension feature different from semantic features, are proposed to measure the knowledge characteristics of questions. The high-dimensional feature vector is obtained by mapping the knowledge representation in questions to the latent knowledge space.

4. The S-KMN model

In this paper, we propose a Semantic-Knowledge Mapping Network (S-KMN) that incorporates semantic features learning and knowledge mapping network for quiz question annotation, as shown in Fig. 3. The S-KMN can be summed up as three main processes: context-aware knowledge representation, knowledge feature extraction based on latent knowledge space, and question annotation with semantic-knowledge features. Designing KMN (Knowledge Mapping Network) to extract implicit knowledge features is the focus of this study. First, knowledge attribute vocabulary is selected from a large number of learning resources from both formal knowledge and practical knowledge. On this basis, the knowledge attribute graph is constructed for each question, which is centered on a question, and the knowledge attribute words and neighbor words are connected to the question. Based on the graph structure, the contextual information of knowledge attribute words is aggregated through "Pyramid Aggregation" to form the knowledge representation of questions. Then, the knowledge representation is projected to latent knowledge space through the learnable projection matrix. Finally, weighted summation is performed on different scene bases in the latent knowledge space to form the knowledge features. To further enrich the feature representation of questions, the vector knowledge attention mechanism is applied to automatically assign adaptive weights to fuse semantic and knowledge features.

4.1. Context-aware knowledge representation

Some words play an important role in conveying knowledge information in question texts. To effectively select these knowledge attribute words, a knowledge attribute vocabulary is established based on a large number of learning resources from both formal knowledge and practical knowledge. This vocabulary includes professional nouns in disciplines such as words, phrases, and symbols. Since the word-level information is limited, the context and sequence information of knowledge attribute words in the question text are also considered in this paper. Hence, each question is represented as a knowledge attribute graph by considering its knowledge attribute words, neighbor words and sequence information. Then, pyramid aggregation as a novel aggregation operation is presented to aggregate the contextual information of knowledge attribute words. Finally, these enhanced knowledge attribute words are arranged in the original order to form the knowledge representation in questions. The whole process is shown in Fig. 4.

4.1.1. Knowledge attribute vocabulary selection

From the two perspectives of formal knowledge and practical knowledge, we extract appropriate knowledge attribute words from a large number of learning resources, including textbooks, supplementary books, after-school homewok, and quiz questions. These two types of knowledge summarize the application scenarios of knowledge points from different dimensions. According to their respective characteristics, TF-IDF and TextRank are applied to extract knowledge attribute words, including a set of words, phrases, and symbols. The specific manifestations are shown in Table 2.

4.1.1.1. The knowledge attribute words of formal knowledge. Textbooks or supplementary books offer the most intuitive explanation of the application scenarios of knowledge points. They are also the standard form of storing, organizing, and expressing domain knowledge. They play an important role in the learning process and belong to formal knowledge (Larkin, 1981). Since they are mainly intuitive explanations of knowledge concepts, words with knowledge characteristics appear more frequently. Therefore, we divide teaching materials and supplementary books into different documents according to application scenarios of knowledge points and use TF-IDF (term frequency-inverse document frequency) to

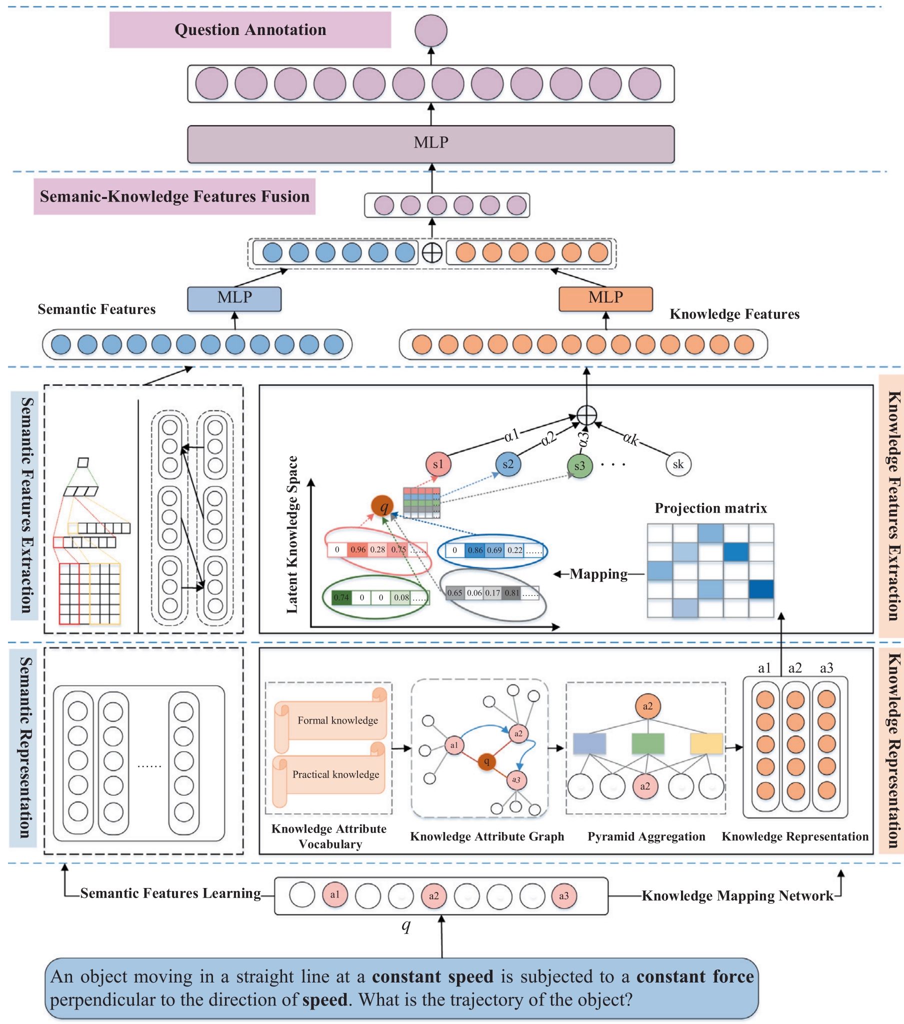

Fig. 3. The overall framework of the proposed S-KMN model. a_i denotes the i-th knowledge attribute word in a question q. The arrows in the knowledge attribute graph represent the sequence information of these words in the question. The different color squares in the "Pyramid Aggregation" represent different receptive fields considered when the knowledge attribute word aggregates neighborhood information. The different colors in the "latent knowledge space" denotes different scene base vectors. α_i represents the weight of the s_i scene base.

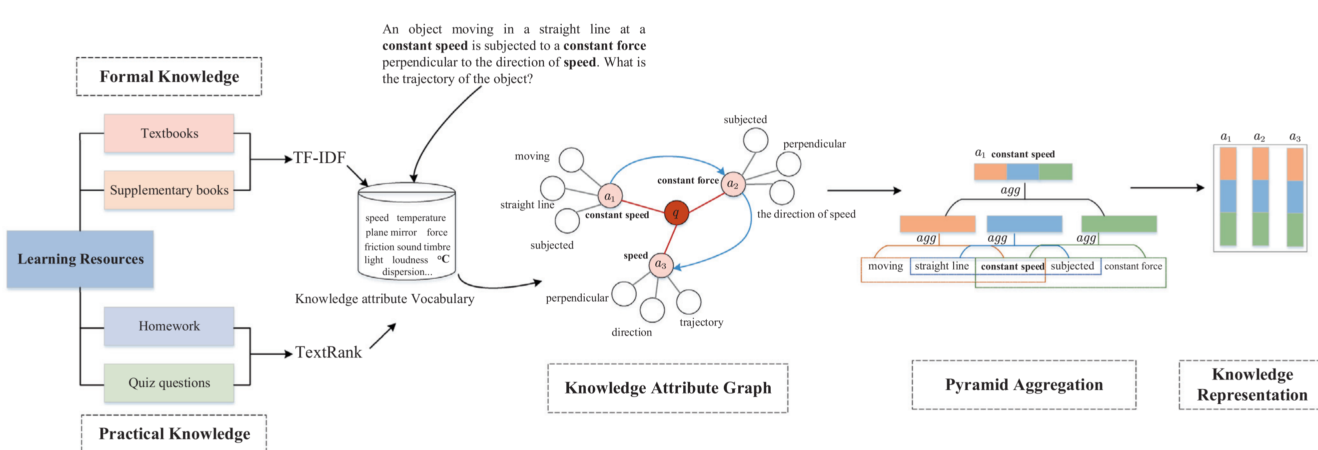

Fig. 4. The process of representing the implicit knowledge information in the question text is mainly divided into the following stages: knowledge attribute vocabulary selection, knowledge attribute graph construction, pyramid aggregation, and the ordering.

Table 2
The main types of knowledge attribute words.

Learning resources	Types of knowledge attribute words
Textbooks or supplementary books (Formal Knowledge)	High-frequency words Unit Symbol
After-school homework or quiz questions (Practical Knowledge)	Co-occurrence words Application scenario words

calculate the TF-IDF factor for each word w^f, thereby selecting the specific words, units, and symbols of knowledge concepts (Tang et al., 2020).

4.1.1.2. The knowledge attribute words of practical knowledge. After-school homework and quiz questions are important tools for students' knowledge diagnosis, and they are flexible applications of professional knowledge in specific scenarios that belong to practical knowledge (Engstrom, 2009). Hence, the high-frequency words in a question are not the most important, while the common words in the application scenarios of knowledge concepts are more meaningful, such as "rainbow" in the "dispersion of light" scenario, and "glasses" and "microscope" in the "lens" scenario. Therefore, TextRank is applied to calculate a weighted score w^t for each word based on a large number of quiz questions and exercises; TextRank is a graph-based ranking algorithm that considers global information rather than rely only on local vertex-specific information (Mihalcea and Tarau, 2004).

4.1.1.3. The combination of knowledge attribute words. To determine the final knowledge attribute vocabulary, the normalization operation is applied to standardize the TF-IDF factor and the TextRank score, and their values are mapped to a certain numerical interval through function transformation. The formula is shown in (1)-(3), where w^f refers to the TF-IDF factor, w^t denotes the TextRank score, and the weight of each word or symbol is mapped to $(0, 1)$ based on the maximum and minimum values. Then, the normalized weight of each word is added to get the importance score w.

$$\widetilde{w^f} = \frac{w^f - w^f_{\min}}{w^f_{\max} - w^f_{\min}} \tag{1}$$

$$\widetilde{w^t} = \frac{w^t - w^t_{\min}}{w^t_{\max} - w^t_{\min}} \tag{2}$$

$$w = \widetilde{w^f} + \widetilde{w^t} \tag{3}$$

The above normalization operation is aimed at the keyword shared by formal knowledge and practical knowledge. For their respective keywords, the corresponding weights continue to maintain their initial values. Finally, the words with higher weights are selected as knowledge attribute words, denoted as $\mathscr{K}^a = \{a_1, a_2, \cdots, a_m\}$, which can best reflect the knowledge information.

4.1.2. Knowledge attribute graph construction

Each question is represented as a graph-like structure (Shin et al., 2019), denoted as $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{O})$, where $\mathscr{V}$ denotes the node set that contains three types of nodes: question, knowledge attribute words appearing in the question and their neighbor words; $\mathscr{E}$ denotes edge set that refer to the relationship between different types of nodes, and $\mathscr{O}$ denotes the order information of knowledge attribute words in the question text. The construction of the knowledge attribute graph is divided into three main stages, as shown in Fig. 4: 1) In the first stage, the position of knowledge attribute words in questions is determined. In the above knowledge attribute vocabulary $\mathscr{K}^a$, we calculate the intersection of the question text $\mathscr{Q}$ and knowledge attribute vocabulary $\mathscr{K}^a$, obtaining word sequence $\mathscr{Q}^a = \{t^a_1, t^a_2, \cdots, t^a_n\}$ and corresponding index sequence $\mathscr{I}^a_q = \{i^a_1, i^a_2, \cdots, i^a_n\}$. 2) In this stage, based on the index $\mathscr{I}^a_q$, the neighbor words of each knowledge attribute word are included in the knowledge attribute graph. 3) Finally, the sequence information of different knowledge attribute words is determined based on their original positional relationship. In this way, the knowledge attribute graph of each question conveys the knowledge information involved in the question text.

4.1.3. Pyramid aggregation

To incorporate the local contextual information of knowledge attribute words, a convolution-like operation acts on the knowledge attribute graph to aggregate the rich neighborhood information of knowledge attribute words (Wang et al., 2016). In this paper, we propose a novel knowledge aggregation operation, called "Pyramid Aggregation", which implements two stacked aggregation operations, as shown in Fig. 4. The purpose of the first aggregation is to expand the receptive field of knowledge attribute words. We take the knowledge attribute words as the center and set up a window matrix to move forward and backward to obtain the neighbor words. The sliding window, which is denoted as $\mathsf{C}_{win} \in \mathbb{R}^p$ with all weights equal to one, is used to capture a sequence of words surrounding knowledge attribute words, and p denotes the length of the sliding window. The length of the sliding window determines the range of contextual information. The second aggregation directly acts on the feature information after the first aggregation. It enriches the knowledge attribute word vector by enhancing their own characteristics and fusing neighborhood information in the receptive field. The realization of these two aggregation operations is similar to (Lin et al., 2020), *agg* incorporates the contextual information into a single vector to enrich the information of knowledge attribute words. The specific formula is defined as follows:

$$\mathbf{Q} \otimes \mathsf{C}_{win} = \left\{ \left[\mathbf{v}^a_{1_1}, \cdots \mathbf{v}^a_{1_p} \right], \cdots, \left[\mathbf{v}^a_{n_1}, \cdots \mathbf{v}^a_{n_p} \right] \right\} \tag{4}$$

$$\mathbf{v}^a_i = agg\left(\mathbf{v}^a_{i_1}, \mathbf{v}^a_{i_2}, \cdots \mathbf{v}^a_{i_p} \right) \tag{5}$$

$$\mathbf{Q}_k = \{ \mathbf{v}^a_1, \mathbf{v}^a_2, \cdots, \mathbf{v}^a_n \} \tag{6}$$

$\otimes$ refers to a window of length p; this sliding window acts in the context of the i-th knowledge attribute word to produce $\left[\mathbf{v}^a_{i_1}, \mathbf{v}^a_{i_2}, \cdots \mathbf{v}^a_{i_p} \right]$ with the p-th times sliding. $\mathbf{v}^a_{i_p}$ denotes the embedding vector of a word. The number of knowledge attribute words in the question determines the number of sliding windows, and the length of the sliding window directly affects the output of the first aggregation. Aggregation operations include the following three main types (Lin et al., 2020). We will evaluate these aggregations in the experimental section.

$$\begin{cases} agg_s = \sigma\left(\Theta \sum_{j \in [1,p]} \mathbf{v}^a_{i_j} + \mathbf{b} \right) \\ agg_m = \sigma\left(\Theta \sum_{j \in [1,p]} \mathbf{v}^a_{i_j} / p + \mathbf{b} \right) \\ agg_c = \sigma\left(\Theta \left(\mathbf{v}^a_{i_1} \| \mathbf{v}^a_{i_2} \| \cdots \mathbf{v}^a_{i_p} \right) + \mathbf{b} \right) \end{cases} \tag{7}$$

where —— denotes the concatenation operation, agg_s nonlinearly transforms the summation of representation vectors, agg_m is to calculate the mean of each dimension in the representation vector and then performs a nonlinear transformation on the resulting vector, and agg_c concatenates the representation vectors before performing the nonlinear transformation. The knowledge attribute word vectors after the aggregation function are arranged in the original order of the question text, and finally, a joint representation $\mathbf{Q}_k$ is generated as the knowledge representation. Since each question contains a different number of knowledge attribute words, we also utilize a zero-padding operation to add each question to the fixed-length input.

4.2. Knowledge feature extraction based on latent knowledge space

Knowledge attribute words have different knowledge understanding in different application scenarios. For example, "speed" can describe not only the motion scene of an object but also the speed of an object under force. The former focuses on the speed of the object, while the latter focuses on the force of an object. To analyze the knowledge expression of knowledge attribute words across multiple scenarios, we create a latent knowledge space with a scene association matrix as the base vector, where the knowledge connotation is more similar between feature vectors with closer distances. Then, a learnable projection matrix maps the knowledge representation in the question text to the latent knowledge space to obtain the vector representation on different scene bases. Finally, the weighted combination of these scene base vectors is the knowledge features of questions. The whole process is shown in Fig. 5.

4.2.1. Latent knowledge space construction

A latent knowledge space with a scene association matrix as a base is constructed to extend the connotation and extension of knowledge points. In this space, the scene association matrix consists of the base vectors of multiple scenarios; this refers to the relative weights of knowledge attribute words in each knowledge scenario. The scene association matrix is the core element of the latent knowledge space and contains rich knowledge connotations. As shown in Fig. 6, each row is a knowledge attribute word, each column represents a type of knowledge scenario, and the intersection refers to the importance of the current word in the application scenario. Relying on the scene association matrix, it can be observed that there are two types of knowledge attribute words. The one type comprises with higher weights but widely applied in multiple knowledge scenarios, such as the word "speed" and "constant speed", which have higher weights in "Speed" and "Force" scenarios. Since these words frequently appear in multiple knowledge scenarios, their weights should be relatively reduced. The other type of knowledge attribute words has lower weights but is relatively independent of the knowledge scenario. For example, the word "Celsius" appears relatively less frequently in the "Temperature" scene. However, the uniqueness of "Celsius" can increase its associated weight with the "Temperature" scene.

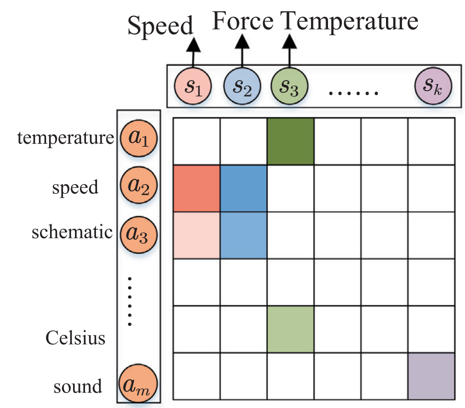

Fig. 6. The scene association matrix. The darker the color of the square is, the more widely the word is used in the corresponding knowledge scenario.

Hence, in addition to considering the weight of knowledge attribute words $\widetilde{w}_{a_i}^{s_j}$ in the current knowledge scenario, it is also necessary to consider the frequency of the word in other knowledge scenarios, to further update the scene association matrix.

Inspired by the relative frequency model (Li et al., 2021), we propose the weight calculation model of knowledge attribute words in different knowledge scenarios. It considers not only the importance of knowledge attribute words in a single knowledge scenario but also the mutual influence between different knowledge scenarios. In the above process of selecting knowledge attribute words, the weight of each knowledge attribute word $\widetilde{w}_{a_i}^{s_j}$ is used as the initial value of the scene association matrix.

For the knowledge attribute words that appear repeatedly in multiple knowledge scenarios, we first sum up their weights in different knowledge scenarios to obtain a sum_{a_i} and then calculate the weight ratio of the knowledge attribute word a_i in the current knowledge scenario s_j as the decay factor $r_{a_i}^{s_j}$. Based on the initial weight w and decay factor $r_{a_i}^{s_j}$, the updated scene association weight can be obtained after multiplication. In contrast, if a knowledge attribute word appears in only one knowledge scenario, then the word is unique to that knowledge scenario, and the word's associated weight will increase accordingly. The calculation of scene association weight is defined as:

$$\mathrm{sum}_{a_i} = \sum_{s_j \in k} w_{s_j}^{a_i} \tag{8}$$

$$r_{a_i}^{k_j} = w_{s_j}^{a_i} / \mathrm{sum}_{a_i} \tag{9}$$

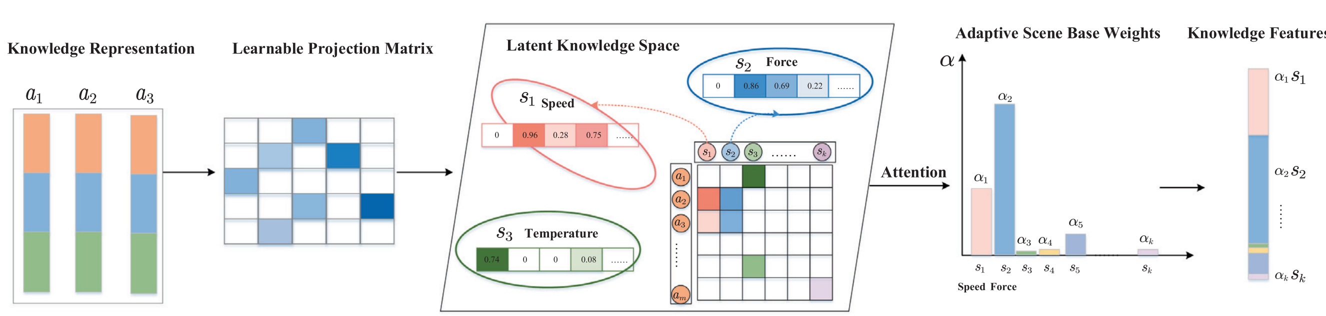

Fig. 5. The process of extracting the implicit knowledge features. It is mainly divided into the following stages: the construction of latent knowledge space with a scene association matrix as the base vector, knowledge mapping, and the weighted combination of scene base vectors.

$$\widetilde{w}_{a_i}^{s_j} = \begin{cases} w_{a_i}^{s_j} \times r_{a_i}^{s_j} & \text{num}(a_i \in s_j) > 1 \\ 1 - w_{a_i}^{s_j} & \text{num}(a_i \in s_j) = 1 \text{ and } w_{a_i}^{s_j} <= 0.5 \\ w_{a_i}^{s_j} & \text{num}(a_i \in s_j) = 1 \text{ and } w_{a_i}^{s_j} > 0.5 \end{cases} \tag{10}$$

where sum_{a_i} denotes the sum of the weights of knowledge attribute words a_i appearing in different knowledge scenarios, $r_{a_i}^{s_j}$ denotes the weight decay factor of the knowledge attribute word a_i in the knowledge scenario s_j, and $\widetilde{w}_{a_i}^{s_j}$ denotes the final updated weight between each knowledge attribute word and different knowledge scenarios. If $\text{num}(a_i \in s_j) = 1$, then a_i exists only in knowledge scenario s_j and the associated weight of a_i should be increased, so we take the larger value between $w_{a_i}^{s_j}$ and $1 - w_{a_i}^{s_j}$. All words are stacked to perform a scene association matrix $A \in (0,1)^{m \times k}$, which is an asymmetric matrix.

4.2.2. Knowledge mapping

The knowledge representation of questions $\mathbf{Q}_k$ is mapped into the d-dimensional latent knowledge space to extract high-level knowledge features. Since the scene base vector is the basic component of the latent knowledge space, the knowledge representation is projected on each scene base through the learnable projection matrix to obtain the corresponding feature representation. Scene base vectors are adaptive for each question in the latent knowledge space, as shown in Fig. 5. According to the knowledge attribute words in the question, a subset of the scene association matrix $\mathbf{A}_q \in \mathbf{A}$ is selected as the base vector. For instance, there are n knowledge attribute words in a question, and the scene base vector corresponding to the question is $A_q \in \mathbb{R}^{n \times k}$. To obtain the mapping vector in each knowledge scene, a projection matrix $\mathbf{H}$ is learned to project the knowledge representation $\mathbf{Q}_k$ to the scene base $\mathbf{A}_q$ in the latent knowledge space. The specific mapping network is defined as follows:

$$\mathbf{Q}_k' = \varphi\left(\mathbf{A}_q^T \mathbf{Q}_k \mathbf{H} + \mathbf{b}\right) \tag{11}$$

where $\mathbf{H} \in \mathbb{R}^{v \times d}$ denotes a trainable projection matrix; v denotes the dimension of the word vector; $\mathbf{A}_q^T$ denotes the transpose of $\mathbf{A}_q$; φ denotes an activation function, e.g, ReLU$\varphi(\mathrm{x}) = \max(0, \mathrm{x})$; $\mathbf{Q}_k' \in \mathbb{R}^{k \times d}$ is the knowledge representation of each question; and d is the dimension of knowledge features.

4.2.3. Knowledge features extraction

The feature representation of each question in the latent knowledge space consists of mapping vectors on all scene bases, but each scene base plays a different role in the knowledge representation of questions. To highlight scene information related to the knowledge connotation of questions, we apply an attention mechanism similar to (Zhou et al., 2016), which automatically assigns the attention score to k scene base vectors of the latent knowledge space. Hence, the final knowledge feature $\mathbf{Q}_k^*$ is formed by a weighted sum of all scene base vectors.

$$\mathbf{Y} = tanh(\mathbf{Q}_k') \tag{12}$$

$$\alpha = softmax\left(\Theta^T \mathbf{Y}\right) \tag{13}$$

$$\gamma = \mathbf{Q}_k' \alpha^T \tag{14}$$

$$\mathbf{Q}_k^* = tanh(\gamma) \tag{15}$$

where $\mathbf{Q}_k'$ denotes the output vector of knowledge mapping layer, Θ is a trained parameter matrix, Θ^T is a transpose, and α denotes the weighted value that corresponds to the output vectors. $\mathbf{Q}_k^* \in \mathbb{R}^d$ is the final knowledge features of questions.

4.3. Question annotation with semantic-knowledge features

To further enrich the feature representation of questions, the integration module fuses the semantic features on the basis of extracted knowledge features. Semantic-knowledge features simulate the thought process of an annotator labeling a quiz question, that is, to analyze the knowledge intention of questions while understanding the text semantics of questions.

4.3.1. Semantic features representation

To capture high-level semantic representations of the question text, each word t_i is embedded into a v-dimensional word vector space to get $\mathbf{v}_i$. The word vectors are learned by Word2Vec based on their local co-occurrences. Wikipedia and educational resources constitute the learning corpus, which considers both the general semantics of the question text as well as its specific semantics in the educational field.

$$\mathbf{v}_i = \mathbf{E} t_i \tag{16}$$

$$\mathbf{Q}_s = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\} \tag{17}$$

where t_i denotes the i-th word of the question text. t_i is represented by a one-hot vector and is then replaced by a vector in the word vector space. $\mathbf{E} \in \mathbb{R}^{f \times v}$ denotes the pre-trained embedding matrix, f denotes the size of the vocabulary, and v is the dimension of the word vector. Hence, a question $\mathcal{Q}$ can be transformed into an embedding matrix $\mathbf{Q}_s \in \mathbb{R}^{n \times v}$ as the semantic feature representation, where n denotes the maximum number of words.

Given an embedding matrix $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\}$, text classification models are always used to extract general semantic features $\mathbf{Q}_s^*$. At present, the dominant semantic feature extraction models mainly include CNNs, RNNs, GNNs, ensemble models and pre-trained language models. Among these models, CNNs, RNNs, GNNs and attention mechanisms have been the most widely used. On this basis, some researchers have presented a reasonable combination of CNNs and RNNs to achieve better results. With the success of pre-trained language models, recent studies have applied them (such as ELMo, GPT, Bert and XLNet) to a variety of downstream tasks in a light fine-tuning way. These methods are described in detail in the related work. To verify the transferability and wide applicability of the KMN proposed in this paper, we conduct a variety of text classification methods, including CNNs, RNNs, GNNs and Bert, to learn semantic features of questions.

4.3.2. Semantic-knowledge features fusion

Inspired by recent advances in vector knowledge attention (Zhan et al., 2020), it has proven to maintain richer information and stronger representation ability than scalar knowledge attention. We apply vector knowledge attention to assign the adaptive learning weight vector to the combination of semantic features and knowledge features automatically, thereby obtaining a comprehensive feature representation (Vaswani et al., 2017; Zahedi et al., 2020). Since the attention score is calculated in the manner of vector, the final question representation learns rich information from the semantic level and the knowledge level by assigning weight to each dimension feature.

The attention calculation procedure can be summarized as follows: First, the semantic feature vector $\mathbf{Q}_s^*$ and the knowledge feature vector $\mathbf{Q}_k^*$ are transformed into $\mathbf{V}_s$ and $\mathbf{V}_k$ through the projection matrices Θ_s, Θ_k respectively, and then they are concatenated into one vector $\mathbf{V}_f$. Next, the nonlinear transformation layers are utilized to calculate the attention weights of semantic features and knowledge features, denoted as att_{score}. Finally, we apply element-wise multiplication to calculate the product of attention vector and the question feature vector, and obtain a rich feature representation that fuses semantic information and knowledge information. The specific formula is as follows:

$$\mathbf{V}_s = \Theta_s \mathbf{Q}_s^* + \mathbf{b}_s \tag{18}$$

$$\mathbf{V}_k = \Theta_k \mathbf{Q}_k^* + \mathbf{b}_k \tag{19}$$

$$\mathbf{V}_f = \mathbf{V}_s || \mathbf{V}_k \tag{20}$$

$$att_{\text{score}} = \varphi(\Theta_{att} \mathbf{V}_f + \mathbf{b}_{\text{att}}) \tag{21}$$

$$\mathbf{Q}_{sk} = att_{\text{score}} \odot [\mathbf{Q}_s^* || \mathbf{Q}_k^*] \tag{22}$$

where Θ_s and Θ_k are the transformation weight matrices, and $\mathbf{b}_s$ and $\mathbf{b}_k$ denote the corresponding bias vectors. —— represents multi-feature concatenation. φ denotes the sigmoid activation function that gives a probabilistically interpretable prediction. $\odot$ denotes the element-wise multiplication of the two matrices.

4.3.3. Question annotation

With the semantic-knowledge features fusion module, $\mathbf{Q}_{sk}$ as the final question representation is used to assign a correct knowledge point y to the input question by a softmax layer. The softmax function normalizes the prediction probability to produce a k-dimensional real number vector, where each element in the vector has a value between $(0, 1)$. Then, the knowledge point with the highest probability is chosen as the predicted label for the input quiz question. The function is formulated as:

$$y = softmax(\Theta \mathbf{Q}_{sk} + \mathbf{b}) \tag{23}$$

For the multi-class classification task, the cross-entropy loss function is adapted to train our model, thus minimizing the following loss (Tao et al., 2019).

$$\text{Loss} = -\sum_{q \in \mathscr{P}} \sum_{i=1}^{k} y_i \log(\hat{y}_i) \tag{24}$$

where $\mathscr{P}$ denotes the training question dataset, $\hat{y}_i$ denotes the prediction probability distribution for the question q, y_i denotes the one-hot represented ground truth of question q, and k denotes the number of knowledge points. The goal of training is to minimize the cross-entropy error between $\hat{y}_i$ and the ground truth y_i for all training data. During training, Adam acts as an optimizer to minimize the loss function (Kingma and Ba, 2015), which is a fast and computationally efficient tool for gradient-based optimization.

5. Experiment

In this section, we conduct comprehensive experiments to evaluate the performance of the Semantic-Knowledge Mapping Network (S-KMN) on question annotation tasks in physics. In what follows, we offer a detailed description of the experimental datasets, baseline methods, and experimental settings, and a specific analysis of the experimental results.

5.1. Physics quiz questions dataset

The data used in this study are a real-world dataset of junior high school physics quiz questions with a total of 19,410 questions covering 30 knowledge points from grades 7 and 8. These knowledge points include temperature, density, mass, measurement, buoyancy, pressure, force, velocity, etc. The dataset is provided by our collaborator, an educational technology company. Each question consists of the question text and the option, along with the corresponding knowledge point. These questions are manually labeled by a team of professional teachers in our partner company. To ensure annotation quality and consistency, three annotators with specialized knowledge of the physical disciplines, participated in the annotation task. The entire annotation proceeds in a two-stage process, where the annotators independently label the entire dataset of 19,410 questions in Stage 1. Each of the questions where the annotators do not have complete agreement is negotiated by all annotators to determine the final knowledge point in Stage 2. This process produces high-quality knowledge points for the physics quiz question dataset.

Some examples from the dataset are shown in Table 3. The dataset contains a variety of question types, such as multiple-choice questions, calculation questions, fill-in-the-blank questions, and experimental questions. The question types and the corresponding quantity distribution are shown in Fig. 7, among which multiple choice questions and fill-in-the-blank questions account for a large proportion. The original questions in our experiment is from physical discipline and is in Chinese, we translate they into English for a better explanation, but our method has no restrictions on the language and discipline.

5.1.1. Dataset statistics

For the dataset, we randomly select 80% as the training set, with the remaining 20% as the test set, and balance the question distribution in different knowledge points. The summary statistics of the training set and the test set are shown in Table 4. A large difference can be seen in terms of the question length and the number of knowledge attribute words due to the datasets containing multiple types of questions. After preprocessing operations, such as word segmentation and removal of stop words, the length of the question is $(0, 150]$, and the number of knowledge attribute words is $(0, 25]$, which accounts for 96%. Therefore, the fixed length of each question is set to 150, and the fixed number of knowledge attribute words is 25. Table 4 also shows that the number of questions without knowledge attribute words in the statistical training set and test set are 247 and 60, respectively, which account for a small proportion. The analysis result can also certify that the selected knowledge attribute words are universal and suitable for most questions. Regardless of the question type, knowledge attribute words reflect their common knowledge information.

5.1.2. Knowledge attribute wordcloud

A total of 435 knowledge attribute words are selected from numerous physics educational resources. The wordcloud of these knowledge attribute words is shown in Fig. 7. The size of a word in the wordcloud is proportional to its importance in the latent knowledge space, and different colors make it visually appealing and readable. Fig. 7 clearly shows that some knowledge attribute words, such as "temperature", "force", and "m/s", belong to the

Table 3
Examples of quiz questions labeled with knowledge points in the experimental data.

Question	Knowledge point
According to the reflected ray ob in the figure, draw the incident ray ao.	reflection of light
What two forces act on a blackboard eraser resting on a horizontal table?	force
Please use physics knowledge to explain why a pressure cooker cooks rice quickly.	pressure
The various colors displayed on the TV screen consist of three colors of light, which are ___.	dispersion of light
For the state change associated with the process of "dropping water into ice" is ___.	change of state

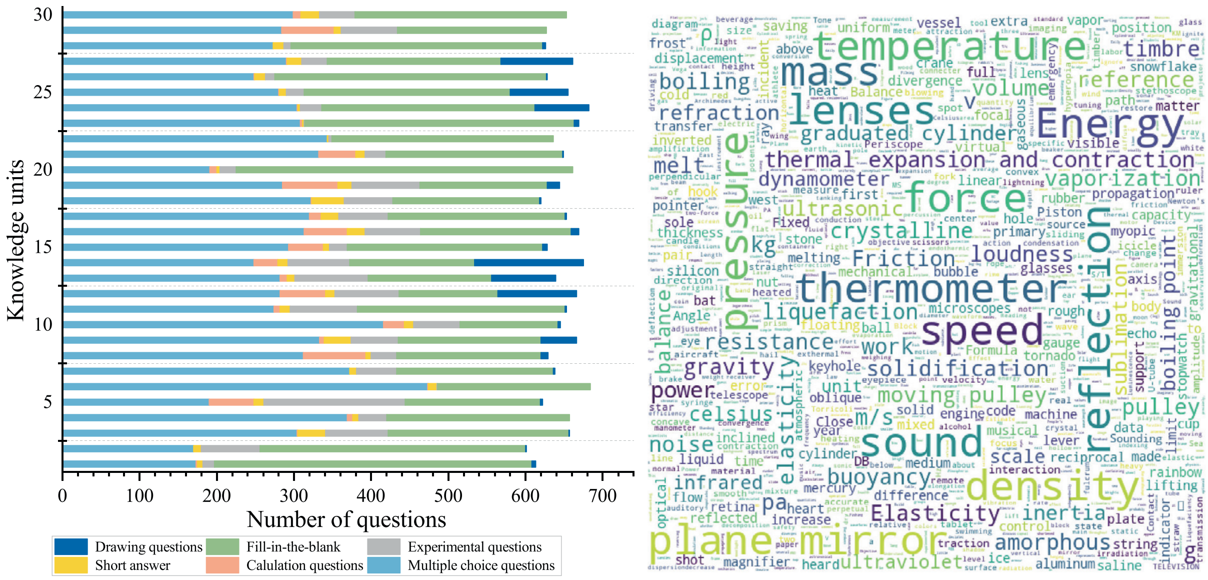

Fig. 7. Left: The statistic of physics question dataset. Right: Wordcloud for the knowledge attribute vocabulary.

Table 4
The summary statistics of the training set and test set.

Data types	Number of questions	Number of words			Number of knowledge attribute words			
		$0 < l \leqslant 10$	$10 < l \leqslant 50$	$50 < l \leqslant 150$	0	$0 < c \leqslant 10$	$10 < c \leqslant 25$	> 25
training data	15527	2747	11543	1237	247	11022	3681	577
Test data	3883	663	2899	321	60	2773	887	163

special terms of physics and play an important role in revealing the connotation and extension of knowledge points. This finding further shows that the knowledge attribute words extracted in this paper have high quality, which is helpful to understand the knowledge information of physical disciplines.

5.2. Experimental settings

5.2.1. Implementation details

During the pre-processing step, a physical dictionary-based Chinese word segmentation tool PkuSeg is applied to the question text. PkuSeg can achieve field-based word segmentation, for example, "States of change" as a whole word instead of being split into multiple words. Based on word segmentation, useless information (such as stop words, punctuation, and number) are removed from the tokenized sequence data. Then, we pre-train the Word2Vec (Mikolov et al., 2013) over the collection of Wikipedia and large-scale education resources to learn the 400-dimensional word embedding, which gives the words in the question both general semantic information and domain information. The words not appearing in the pre-trained vocabulary are randomly initialized with uniformly distributed values between −0. 25 and 0.25.

For the hyper-parameters setting in our model, KMN with hidden layer dimensions of 256 applies Rectified Linear Units (RELU) as the activation function with a drop rate of 0.3. As an aggregation function, $aggre_m$ is used to enrich the contextual information of knowledge attribute words. To avoid over-fitting, dropout at a rate of 0.5 is applied after the fully connected layer. In the training procedure, the gradient-based Adam optimizer (Kingma and Ba, 2015) has an initial learning rate of 5^{-3}. Our method is trained at a batch size of 32 and stops training when the accuracy on the validation set does not increase for 20 consecutive epochs. For all baselines, we use default parameter settings as in their original papers or implementations.

5.2.2. Evaluation metrics

Concerning the evaluation methods, precision, recall, and the F1-score have frequently been used as evaluation metrics in binary classification tasks. Accuracy is a relatively simple evaluation method that refers to the ratio of the number of correctly classified questions to the total number of questions. To further analyze the distribution of predicted labels and real labels in different categories, precision and recall are proposed. Precision is defined as the ratio of true positive (TP) samples in the all-positive predicted samples. Recall is defined as the ratio of true positive samples to all samples with correct predictions. Since precision and recall are a pair of contradictory measures, F1 is induced to the symmetrical average of two evaluation metrics. The calculation formulas of these three evaluation metrics are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{25}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{26}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{27}$$

In a multi-class scenario, macro-averaging is used to summarize the results of precision, recall and F1 computed for all knowledge points. The macro-averaging strategy performs an average over the evaluation measure considering all classes and is not affected by data imbalance. Therefore, the formulas for precision, recall and F1 by using the macro-averaging strategy are as follows.

$$\text{Macro} - \text{Precision} = \frac{1}{k}\sum_{i=1}^{k}\text{Precision}_i \quad (28)$$

$$\text{Macro} - \text{Recall} = \frac{1}{k}\sum_{i=1}^{k}\text{Recall}_i \quad (29)$$

$$\text{Macro} - \text{F1} = \frac{1}{k}\sum_{i=1}^{k}F1_i \quad (30)$$

5.2.3. Baseline models

Since question annotation belongs to a special text classification task, some state-of-the-art text classification models for extracting semantic features are selected as baseline models. All models are fine-tuned on the physical quiz question set to demonstrate the effectiveness of the S-KMN. The relevant introduction and hyper-parameters of these models are as follows:

- **TextCNN**: It is proposed by Kim (Kim, 2014), which is a variant of CNN superior in understanding semantics. The innovation of the model is that the convolution layer has three different kernels. The filter windows of 3, 4, 5 with 100 feature maps each are applied in this paper.
- **Att-BLSTM** (Zhou et al., 2016): This is a classifier with a BLSTM network and a neural attention mechanism that can capture the most important semantic information in the text. The architecture comprises five components, namely, the input layer, embedding layer, LSTM layer, attention layer, and output layer. The single LSTM block uses 256 dimensional hidden and cell states.
- **AC-BiLSTM** (Liu and Guo, 2019): This is a hybrid model architecture that contains a BiLSTM block, an attention mechanism and a convolutional layer. It captures both the local features of phrases and the global semantics of sentences. The number of neurons in a single LSTM is set to 256, the filter size in the convolutional layer is 3, and the number is set to 100.
- **Attention-based BiGRU-CNN (ABBC)** (Liu et al., 2019): This neural network integrates the advantages of Text-CNN, Bi-GRU and an attention mechanism for Chinese question classification. The model extracts the key features while extracting the contextual information of the words in each question. The height of the convolution kernel is 2, 3, and 4; and hidden layer dimension is 256.
- **TextGCN** (Yao et al., 2019b): In this paper, graph convolutional networks are used to construct a text graph based on question datasets, which capture word co-occurrence and question word relations. This is the first study to jointly learn word and document embeddings through the construction of heterogeneous graphs. The embedding size of the first graph convolution layer is 200 and the window size is set to 10 due to the shorter question text.
- **BERT** (Devlin et al., 2019): BERT, as a language representation model, is designed to learn text representations by using masked language models and a "next sentence prediction" task. $Bert_{Base}$ with 12 transformer layers, 768 hidden units, and 12 multi-attention heads is pretrained on Chinese Wikipedia data. We fine-tuned the final classification layer on our quiz question annotation task.
- **TextING** (Zhang et al., 2020b): This model builds individual graphs for each sample to depict detailed word-word relations, which learns the word embedding via Gated Graph Neural Networks (Li et al., 2019), and finally aggregates multiple words into the document embedding.
- **ChineseBERT** (Sun et al., 2021): To consider the uniqueness of the Chinese language, ChineseBERT incorporates the glyph and pinyin information of Chinese characters into the pre-trained Bert model. It takes full advantage of the semantics behind the glyphs and addresses the challenge of heteronym.
- **S-KMN (Att-BLSTM):** This is a specific implementation of the proposed S-KMN framework, Att-BLSTM, as a semantic feature extraction network, fused with KMN for quiz question annotation tasks.
- **S-KMN (AC-BiLSTM):** AC-BiLSTM, as a hybrid model integrating the advantages of multiple networks, is chosen as a semantic feature extraction module of S-KMN.
- **S-KMN (Bert):** Bert, as a pre-trained language representation model integrating the context-sensitive features, is chosen as a semantic feature extraction module of the S-KMN.

5.3. Results

5.3.1. Overall comparison

In this subsection, we compare the proposed three S-KMN models with the eight deep learning models in terms of macro-precision, macro-recall and the macro-F1 on real-world physics quiz question sets. The experimental results are summarized in Table 5. The best value is in bold, and the second-best value is underlined. The table shows that the S-KMN incorporating knowledge features outperforms the state-of-the-art text classification-based question annotation method. Table 5 provides the following observations:

- Among all baseline models, ChineseBERT achieves the best results, Bert, AC-BiLSTM and ABBC outperform the single network models TextCNN and Att-BLSTM. This is because ChineseBert considers the glyph and pinyin information of Chinese characters and is thus more suitable for the Chinese question dataset used in this paper. Table 5 also shows that TextING and TextGCN achieve satisfactory results on document classification tasks, but the constructed text graph is not suitable for modeling the relationship between word nodes and question nodes due to the short question text and sparse semantics, thus resulting in poor performance on question annotation tasks.
- The macro-F1 of S-KMN (Att-BLSTM), S-KMN (AC-BiLSTM) and S-KMN (BERT) after knowledge features enhancement improve by $1.21\%, 1.18\%, 1.13\%$, respectively. Since the semantic feature extraction module is the backbone with the same evaluation procedure, the gain is entirely due to the newly introduced knowledge features. The major reason is that the extracted knowledge features capture the implicit knowledge information of questions from a novel feature dimension, which is not considered in these comparison methods.
- The results obtained by the proposed three S-KMN models, shows that the KMN, as a knowledge feature extraction method, can be superimposed on any network that extracts semantic features to form the S-KMN. Therefore, the S-KMN has great compatibility and strong transferability.

Table 5
Performance of the S-KMN against all baseline models.

Methods	Macro-P	Macro-R	Macro-F1
TextCNN	0.8236	0.8201	0.8196
Att-BLSTM	0.8362	0.8329	0.8322
ABBC	0.8401	0.8374	0.8369
AC-BiLSTM	0.8426	0.8397	0.8392
TextGCN	0.8222	0.8212	0.8217
BERT	0.8463	0.8431	0.8426
TextING	0.8367	0.8354	0.8351
ChineseBERT	0.8507	0.8479	0.8482
S-KMN(Att-BLSTM)	0.8471	0.8439	0.8443
S-KMN(AC-BiLSTM)	0.8528	0.8506	0.8510
S-KMN(BERT)	**0.8569**	**0.8533**	**0.8539**

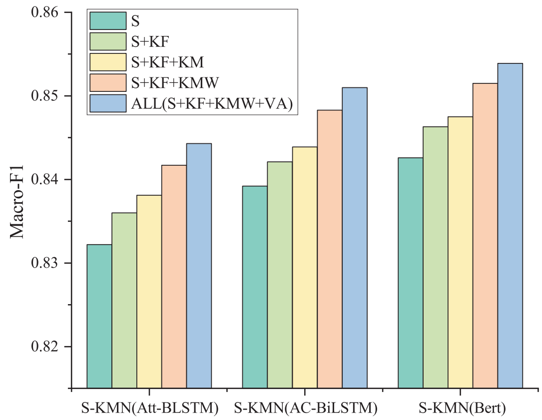

Fig. 8. The ablation study on the S-KMN. "S" denotes the semantic features extraction, "KF" denotes the knowledge representation, "KM" denotes the knowledge mapping, and "VA" denotes the vector knowledge attention mechanism.

5.3.2. Ablation study on the S-KMN

To clarify the effectiveness of several key factors in the S-KMN, a series of ablation studies are conducted on the physics question dataset. Mainly focus on the following three core components of the S-KMN: knowledge representation module, knowledge mapping module, and semantic-knowledge features fusion module. As shown in Fig. 8, "ALL" denotes the entire S-KMN with all components, including semantic features extraction (S), knowledge representation (VH), knowledge mapping (KM), and vector knowledge attention mechanism (VA).

5.3.2.1. Effects of the knowledge representation. The knowledge representation of each question is obtained by enhanced knowledge attribute words and their contextual information based on the corresponding knowledge attribute graph. To verify the effectiveness of the knowledge representation, we compare the performance changes before and after fusing the knowledge representation based on the extracted semantic features. From Fig. 8, we can observe that fusing the knowledge representation slightly increased by 0.38%, 0.29% and 0.36% in macro-F1 compared with the semantic features extracted by Att-BLSTM, AC-BiLSTM and Bert. The possible reason for the performance improvement is that the knowledge representation aggregates knowledge attribute words and their contextual information, and contains rich knowledge information.

5.3.2.2. Effects of the knowledge mapping. The scene association matrix, as the base vector of the latent knowledge space, plays an important role in the knowledge mapping layer. Hence, we investigate the effect of different weights of scene association matrix on model performance. The results are shown in Fig. 8. The baseline is to assign the same weight to all of the knowledge attribute words belonging to each knowledge scenario. In the matrix, if the word appears in a knowledge scene, the value is set to 1, otherwise, the value is 0. This constant-valued scene association matrix can be denoted by KM. Correspondingly, our proposed scene association matrix considers that the same knowledge attribute word has different association weights for different knowledge scenarios (KMW). The experimental results show that compared with the fixed weight, the weighted value achieves stable performance across these three S-KMN variants. The reason could be that the weighted scene association matrix considers the importance of knowledge attribute words in different knowledge scenarios, thereby adding more meaningful knowledge information.

5.3.2.3. Effects of the vector knowledge attention mechanism. The vector knowledge attention mechanism is applied to obtain a weighted sum representation of semantic features and knowledge features instead of direct concatenation. Fig. 8 shows that compared with the concatenation operation, adding a vector knowledge attention layer achieves superior performance across all the baseline models, regardless of the weighted value or the constant value in the scene association matrix. The possible explanation is that adaptively assigning weights to each dimension feature maintains richer information. Hence, the above ablation studies prove that the vector knowledge attention and the weighted scene association matrix are essential to enhance the feature representation of questions in the proposed S-KMN framework.

5.3.3. Results on question types

Most existing question annotation models are only suitable for a few question types. In contrast, the knowledge features proposed in this work directly correspond to knowledge points, and there are no restrictions on question types. The experimental results shown

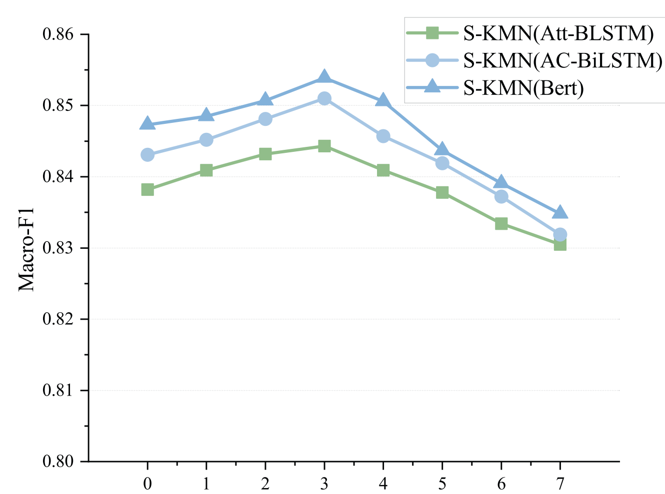

Fig. 9. Effects of different sliding window lengths.

Table 6
The comparison of Macro-F1 on different question types using S-KMN.

Model	Multiple choice questions		Fill-in-the-blank		All question types	
	Macro-F1	Δ	Macro-F1	Δ	Macro-F1	Δ
Att-BLSTM	0.8159	1.26%	0.7903	1.37%	0.8322	1.21%
+KMN	0.8285		0.8040		0.8443	
AC-BiLSTM	0.8205	1.24%	0.8052	1.40%	0.8392	1.18%
+KMN	0.8329		0.8192		0.8510	
Bert	0.8269	1.19%	0.8126	1.32%	0.8426	1.13%
+KMN	0.8388		0.8258		0.8539	

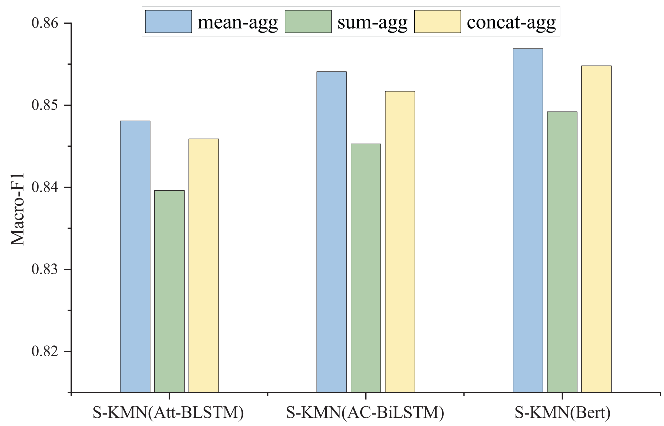

Fig. 10. Effects of different aggregation methods.

in Table 6 verify this hypothesis. Since multiple choice questions and fill-in-the-blank questions account for a relatively high proportion of the dataset, there are enough data to train all models. Therefore, the above-mentioned dataset is divided into three parts: a multiple choice dataset with 8,883 questions, a fill-in-the-blank question dataset with 7,266 questions, and the whole dataset. The ratio of the training set to the test set is 8:2. Experiments are conducted to evaluate the performance of the S-KMN based on Att-BLSTM, AC-BiLSTM, and BERT, which are selected as representatives of different model types over the above three datasets.

The results, as shown in Table 6, illustrate the performance comparison of text classification models and the S-KMN on different question types. The results show that the three baseline models perform outstandingly after integrating the knowledge features. For instance, compared to BERT, S-KMN (BERT) shows an apparent improvement in macro-F1 on the multiple-choice question set, the fill-in-the-blank question set, and the whole dataset of $1.19\%, 1.32\%$, and 1.13%, respectively. Hence, the experimental results show that the growth rate of the S-KMN on the three data sets is significant, thus indicating that our model is not limited by the question type and is applicable to all question datasets.

5.3.4. Hyperparameter research

5.3.4.1. Effects of sliding window length. To investigate the influence of sliding window length, we conduct several experiments by selecting sliding windows of different lengths in the process of aggregating the contextual information of knowledge attribute words. As shown in Fig. 9, the x-axis represents the length of the sliding window, and the y-axis represents macro-F1. The experimental results show that the S-KMN performs better when the length of sliding window is equal to 3. The macro-F1 decreases when the length of the sliding window is too long or too short. The key explanation for this phenomenon is that an excessively short sliding window fails to fully capture the contextual information of knowledge attribute words, and an excessively long sliding window means the repetition and superposition of surrounding words, adding additional interference information.

5.3.4.2. Effects of different aggregators. The sum aggregator, mean aggregator, and concatenation aggregator are three types of aggregators utilized in the S-KMN models, which reflect different aggregating ways of knowledge attribute words integrating surrounding contextual information. As a result, experiments are carried out to see how different aggregators affect the S-KMNs' performance. Fig. 10 shows the macro-F1 score for the quiz question datasets using different aggregators. The figure shows that the sum aggregator and concatenation aggregator produce quite comparable results, but the mean aggregator achieves better experimental results. The reason for this could be that the feature information obtained by the sliding window has certain repeatability, and the mean aggregation makes the final representation of questions strengthen the important feature while reducing the feature dimension.

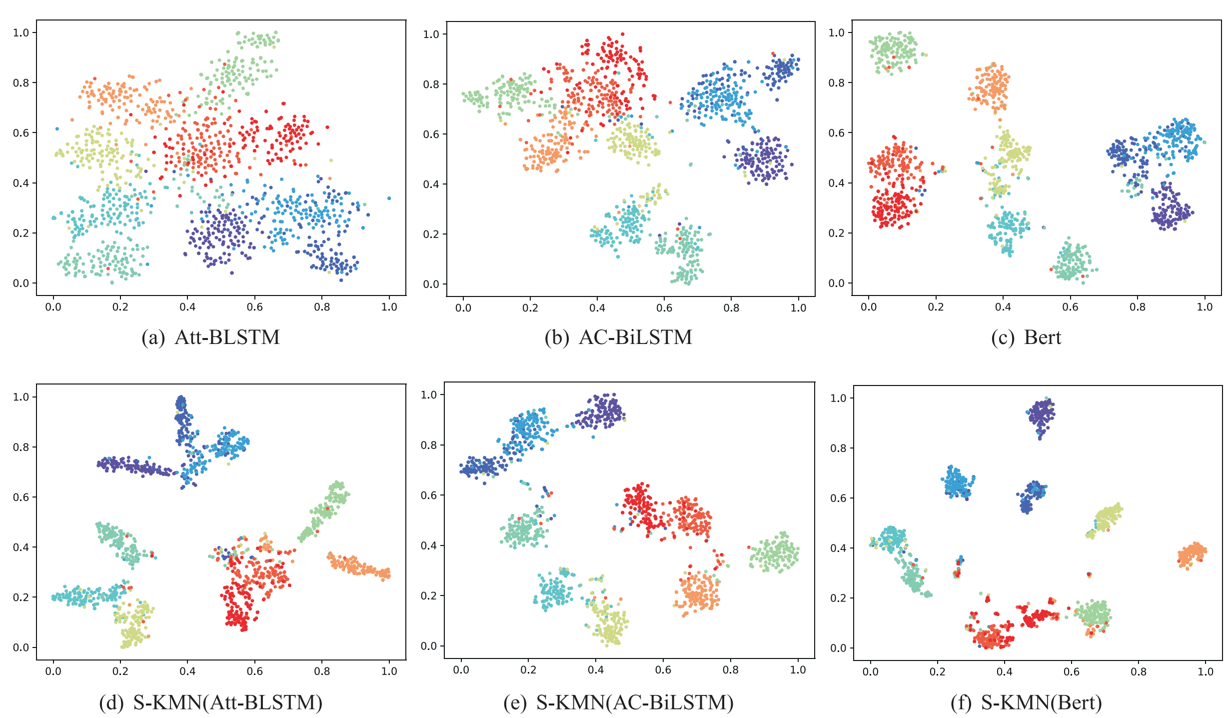

Fig. 11. t-SNE Plots of question feature representation obtained from randomly selecting 10 knowledge points. Different colors represent different knowledge points.

5.3.5. Visualization of question representation

In this subsection, we verify that the knowledge features extracted by the KMN enrich the feature representation of questions. t-SNE is performed to visualize the question representation before and after the fusion of knowledge features. First, we output the question vector of the feature extraction layer from the baseline model and the S-KMN model, that is, Q_s^* and Q_{sk}, which directly determine the accuracy of subsequent question annotation. Since Q_s^* and Q_{sk} are both high-dimensional vector representations, we apply a t-SNE (Van der Maaten and Hinton, 2012) to transform high-dimensional vector into a two-dimensional space to visualize the question representation. As shown in Fig. 11, we randomly select 10 of the 30 knowledge points and visually display the feature representation of questions belonging to these knowledge points.

Fig. 11(a)(b)(c) show that Att-BLSTM, AC-BiLSTM and Bert extract only semantic features of the question text, thus blurring the boundaries of the question feature under different knowledge points. This is prone to cause misjudgment, and the improvement of annotation accuracy can be limited. In contrary, the question features enhanced by superimposing knowledge features are more discriminative, and it is easier to distinguish the questions belonging to different knowledge points, as shown in Fig. 11(d)(e)(f). Fig. 11 also reasonably explains why BERT outperforms Att-BLSTM and AC-BiLSTM.

6. Conclusion and future work

In this paper, a Semantic-Knowledge Mapping Network (S-KMN) is proposed to simulate the process of an annotator thinking about a question, which further mines the knowledge connotation on the basis of understanding the explicit semantics of a question. Designing the KMN to extract knowledge features is the focus of this paper. First, for each question, we construct a knowledge attribute graph, which captures the relationship among a question, knowledge attribute words, and their neighborhood to form the knowledge representation of questions. Then, the knowledge representation is mapped to a latent knowledge space through a learnable projection matrix for extracting knowledge features. To further enrich question representation, the vector knowledge attention mechanism is applied to assign the attentive weights to the combination of explicit semantic features and implicit knowledge features. The effectiveness of the S-KMN framework, especially the important role of the extracted knowledge features, is verified by experiments on the physics quiz question set.

In the future, we will further study the generalization ability of S-KMN in other disciplines and question types, as well as the distinction and connection between knowledge and semantics in question representation. In another promising direction is to comprehensively consider the multimodal information in the question, including text modality and image modality, which reflect the knowledge connotation of the question from multiple perspectives, thereby further enriching the feature representation of quiz questions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No.62177020), Humanities and Social Sciences Youth Foundation, Ministry of Education of the People's Republic of China (No.19YJC880032), Special Research Project on Teacher Education of CCNU (No. CCNUTEIII 2021-07) and Teaching Innovation Research Project of CCNU (No. ZNXBJY202115).

References

Abdous, M., He, W., Yen, C.-J., 2012. Using data mining for predicting relationships between online question theme and final grade. Educational Technology & Society, pp. 77–88.

Ahmad, S., Asghar, M.Z., Alotaibi, F.M., Al-Otaibi, Y.D., 2022. A hybrid cnn+bilstm deep learning-based dss for efficient prediction of judicial case decisions. Expert Syst. Appl. 118318.

Ahmed, I.A., AL-Aswadi, F.N., Noaman, K.M., Alma'aitah, W.Z., 2022. Arabic knowledge graph construction: A close look in the present and into the future. J. King Saud Univ.- Comput. Informat. Sci. 34, 6505–6523.

Almuzaini, H.A., Azmi, A.M., 2022. An unsupervised annotation of arabic texts using multi-label topic modeling and genetic algorithm. Expert Syst. Appl. 203, 117384.

Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., Sima'an, K., 2017. Graph convolutional encoders for syntax-aware neural machine translation. In: Palmer, M., Hwa, R., Riedel, S. (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, pp. 1957–1967.

Chen, H., Wu, L., Chen, J., Lu, W., Ding, J., 2022. A comparative study of automated legal text classification using random forests and deep learning. Informat. Process. Manage., 102798

Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, pp. 4171–4186.

Engstrom, S.P., 2009. The form of practical knowledge. Harvard University Press.

Ethayarajh, K., 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, pp. 55–65.

Guo, B., Zhang, C., Liu, J., Ma, X., 2019. Improving text classification with weighted word embeddings via a multi-channel textcnn model. Neurocomputing 363, 366–374.

Hassani, H., Ershadi, M.J., & Mohebi, A. (2022). Lvtia: A new method for keyphrase extraction from scientific video lectures. Informat. Process. Manage 102802.

Huang, Q., Mo, L., Li, P., Cai, Y., Liu, Q., Wei, J., Li, 2021. Story ending generation with multi-level graph convolutional networks over dependency trees. In: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2021 vol. 14, pp. 13073–13081.

Jang, B., Kim, M., Harerimana, G., Kang, S.-U., Kim, J.W., 2020. Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. Appl. Sci. 5841.

Jia, C., Liang, X., Zhang, Y., 2019. Cross-domain ner using cross-domain language modeling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2464–2474.

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T., 2017. Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics(Volume 2, Short Papers), pp. 427–431.

Kadhim, A.I., 2019. Survey on supervised machine learning techniques for automatic text classification. Artif. Intell. Rev. 52, 273–292.

Kalchbrenner, N., Grefenstette, E., Blunsom, P., 2014. A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, (Volume 1: Long Papers), pp. 655–665.

Khot, T., Clark, P., Guerquin, M., Jansen, P., Sabharwal, A., 2020. Qasc: A dataset for question answering via sentence composition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8082–8090.

Kim, Y., 2014. Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, pp. 1746–1751.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations. ICLR.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., Al-Emari, S., 2020. A systematic review of automatic question generation for educational purposes. Int. J. Artif. Intell. Educ., 121–204

Larkin, J.H., 1981. Enriching formal knowledge: A model for learning to solve textbook physics problems. Cognitive Skills Acquisit. 311–334.

Lee, J., Lee, I., Kang, J., 2019. Self-attention graph pooling. In: International Conference on Machine Learning. PMLR, pp. 3734–3743.

Li, B., Du, X., Chen, M., 2020a. Cross-language question retrieval with multi-layer representation and layer-wise adversary. Inf. Sci. 527, 241–252.
Li, H., Wei, H., Wang, Y., Song, Y., Qu, H., 2020b. Peer-inspired student performance prediction in interactive online question pools with graph neural network. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2589–2596.
Li, K., Feng, Y., Gao, Y., Qiu, J., 2020c. Hierarchical graph attention networks for semi-supervised node classification. Appl. Intell. 50, 3441–3451.
Li, X., Cui, M., Li, J., Bai, R., Lu, Z., Aickelin, U., 2021. A hybrid medical text classification framework: Integrating attentive rule construction and neural network. Neurocomputing, 345–355.
Li, Z., Cui, Z., Wu, S., Zhang, X., Wang, L., 2019. Fi-gnn: Modeling feature interactions via graph neural networks for CTR prediction. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, ACM. pp. 539–548.
Liao, S., Wang, J., Yu, R., Sato, K., Cheng, Z., 2017. Cnn for situations understanding based on sentiment analysis of twitter data. Proc. Comput. Sci. 111, 376–381.
Lin, X., Quan, Z., Wang, Z.-J., Ma, T., Zeng, X., 2020. Kgnn: Knowledge graph neural network for drug-drug interaction prediction. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI, pp. 2739–2745.
Liu, G., Guo, J., 2019. Bidirectional lstm with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338.
Liu, J., Yang, Y., Lv, S., Wang, J., Chen, H., 2019. Attention-based bigru-cnn for chinese question classification. J. Ambient Intell. Humanized Comput., 1–12
Liu, S., Yu, J., Li, Q., Liang, R., Zhang, Y., Shen, X., Sun, J., 2022. Ability boosted knowledge tracing. Informat. Sci. 567–587.
Liu, Y., Yi, X., Chen, R., Zhai, Z., Gu, J., 2018. Feature extraction based on information gain and sequential pattern for english question classification. IET Software 12, 520–526.
Lv, S., Guo, D., Xu, J., Tang, D., Duan, N., Gong, M., Shou, L., Jiang, D., Cao, G., Hu, S., 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8449–8456.
Van der Maaten, L., Hinton, G., 2012. Visualizing non-metric similarities in multiple maps. Machine Learn. 33–55.
Meškelė, D., Frasincar, F., 2020. Aldonar: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. Informat. Process. Manage. 57, 102211.
Mihalcea, R., Tarau, P., 2004. Textrank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119.
Mohasseb, A., Bader-El-Den, M., Cocea, M., 2018. Question categorization and classification using grammar based approach. Informat. Process. Manage. 54, 1228–1243.
Naseem, U., Razzak, I., Khan, S.K., Prasad, M., 2021. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. Trans. Asian Low-Resource Language Informat. Process. 20, 1–35.
Paramasivam, A., Nirmala, S.J., 2022. A survey on textual entailment based question answering. J. King Saud Univ.- Comput. Informat. Sci. 34, 9644–9653.
Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, pp. 1532–1543.
Qiao, C., Hu, X., 2019. Text classification for cognitive domains: A case using lexical, syntactic and semantic features. J. Informat. Sci. 45, 516–528.
Qin, S., Wu, H., Nie, R., He, J., 2020. Deep model with neighborhood-awareness for text tagging. Knowledge-Based Syst. 105750.
Shin, S., Jin, X., Jung, J., Lee, K., 2019. Predicate constraints based question answering over knowledge graph. Informat. Process. Manage. 445–462.
Silva, V.A., Bittencourt, I.I., Maldonado, J.C., 2018. Automatic question classifiers: A systematic review. IEEE Trans. Learn. Technol. 12, 485–502.
Sun, B., Zhu, Y., Xiao, Y., Xiao, R., Wei, Y., 2018. Automatic question tagging with deep neural networks. IEEE Trans. Learn. Technol. 12, 29–43.
Sun, Z., Li, X., Sun, X., 2021. ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2065–2075.
Tang, Z., Li, W., Li, Y., Zhao, W., Li, S., 2020. Several alternative term weighting methods for text representation and classification. Knowledge-Based Syst. 106399.
Tao, H., Tong, S., Zhao, H., Xu, T., Jin, B., Liu, Q., 2019. A radical-aware attention-based model for chinese text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence 01, pp. 5125–5132.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008.
Wang, D., Nyberg, E., 2015. A long short-term memory model for answer sentence selection in question answering. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 707–712.
Wang, J., Wang, Z., Zhang, D., Yan, J., 2017. Combining knowledge with deep convolutional neural networks for short text classification. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 2915–2921.
Wang, P., Xu, B., Xu, J., Tian, G., Liu, C., Hao, H., 2016. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing, 806–814.
Wang, S.I., Manning, C.D., 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 90–94.
Wang, X., Ma, W., Guo, L., Jiang, H., Liu, F., Xu, C., 2022. Hgnn: Hyperedge-based graph neural network for mooc course recommendation. Informat. Process. Manage. 102938.
Wu, Y., Li, J., Wu, J., Chang, J., 2020a. Siamese capsule networks with global and local features for text classification. Neurocomputing 390, 88–98.
Wu, Z., He, T., Mao, C., Huang, C., 2020b. Exam paper generation based on performance prediction of student group. Informat. Sci. 72–90.
Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020c. A comprehensive survey on graph neural networks. IEEE Trans. Neural Networks Learning Syst. 32, 4–24.
Xu, K., Lai, Y., Feng, Y., Wang, Z., 2019. Enhancing key-value memory neural networks for knowledge based question answering. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2937–2947.
Yang, M., Qu, Q., Chen, X., Guo, C., Shen, Y., Lei, K., 2018. Feature-enhanced attention network for target-dependent sentiment classification. Neurocomputing 307, 91–97.
Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS 2019, pp. 5754–5764.
Yang, Z., Liu, Z., Liu, S., Min, L., Meng, W., 2014. Adaptive multi-view selection for semi-supervised emotion recognition of posts in online student community. Neurocomputing 144, 138–150.
Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E., 2016. Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489.
Yao, L., Mao, C., Luo, Y., 2019a. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. BMC Medical Informat. Decis. Mak. 19, 31–39.
Yao, L., Mao, C., Luo, Y., 2019b. Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence 01, pp. 7370–7377.
Ye, H., Yan, Z., Luo, Z., Chao, W., 2017. Dependency-tree based convolutional neural networks for aspect term extraction. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 350–362.
Yilmaz, T., Ozcan, R., Altingovde, I.S., Ulusoy, Ö., 2019. Improving educational web search for question-like queries through subject classification. Informat. Process. Manage. 228–246).
Yin, W., Schütze, H., Xiang, B., Zhou, B., 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Trans. Assoc. Comput. Linguist. 4, 259–272.
Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J., 2018. Hierarchical graph representation learning with differentiable pooling. Adv. Neural Informati. Process. Syst. 31.
Zahedi, M.S., Rahgozar, M., Zoroofi, R.A., 2020. Hca: Hierarchical compare aggregate model for question retrieval in community question answering. Informat. Process. Manage. 102318.
Zeng, H., Zhi, Z., Liu, J., Wei, B., 2021. Improving paragraph-level question generation with extended answer network and uncertainty-aware beam search. Inf. Sci. 571, 50–64.
Zhan, Z., Hou, Z., Yang, Q., Zhao, J., Zhang, Y., Hu, C., 2020. Knowledge attention sandwich neural network for text classification. Neurocomputing, 1–11.
Zhang, M., Chen, Y., 2018. Link prediction based on graph neural networks. Adv. Neural Informat. Process. Syst. 31.
Zhang, Y., Dai, H., Kozareva, Z., Smola, A.J., Song, L., 2018. Variational reasoning for question answering with knowledge graph. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 6069–6076.
Zhang, Y., Dai, H., Yun, Y., Liu, S., Lan, A., Shang, X., 2020a. Meta-knowledge dictionary learning on 1-bit response data for student knowledge diagnosis. Knowledge-Based Syst. 106290.
Zhang, Y., Yu, X., Cui, Z., Wu, S., Wen, Z., Wang, L. 2020b. Every document owns its structure: Inductive text classification via graph neural networks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 334–339.

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., Zhou, X., 2020c. Semantics-aware bert for language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence 05, pp. 9628–9635.
Zhao, H., Liu, Z., Yao, X., Yang, Q., 2021. A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach. Informat. Process. Manage. 58, 102656.
Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., 2020. Graph neural networks: A review of methods and applications. AI Open 1, 57–81.
Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B., 2016. Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 2: Short papers), pp. 207–212.
Zhou, Y., Huang, C., Hu, Q., Zhu, J., Tang, Y., 2018. Personalized learning full-path recommendation model based on lstm neural networks. Inf. Sci. 444, 135–152.