Research on High School Math Exercise Recommendation Based on Graph Neural Network

Wangzhihui Mei

Supervisor: Zhifeng Wang

This thesis is submitted for the degree of

Master of Computer Science

Central China Normal University Wollongong Joint Institute

Central China Normal University

March 2021

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgments.

Wangzhihui Mei
March 2021

Abstract

After 2010, artificial intelligence technology has gradually become a research hotspot in the field of computer technology. In particular, the advent of AlphaGo has aroused great concern in the industry for the prospect of artificial intelligence. This has brought about the explosive development of the industry, and also raised a large number of research topics. In artificial intelligence-related research, various algorithm innovations, theoretical breakthroughs and model applications are emerging one after another, laying the foundation for the intelligence of various industries. The application of artificial intelligence technology in the field of education has also given birth to the emergence of the concept of intelligent education. Among them, adaptive learning is one of the popular application fields in intelligent education[**?**]. Adaptive learning models generally track the learning status of students by combining big data analysis of massive student group learning data and precise data analysis of target student individual data, targeting on personalizing the learning path according to the individual characteristics of the students and the proficiency of knowledge mastery[**?**]. Adaptive learning technology can use automated machine learning algorithms to complete student evaluation and teaching plans that required a lot of manual labor in the past, which can systematically alleviate the current scarcity and uneven distribution of domestic educational resources as well as reduce the burden on education practitioners and students. It also has great development prospects and commercial value. There are more and more artificial intelligence research teams and intelligent education technology companies on the market focusing on the development and application of adaptive learning tools. Some smart educational technology companies have used adaptive learning as the core function or main selling point of their products. Adaptive educational technology can comprehensively analyze students' individual level learning ability, knowledge proficiency and group level popular learning resources, error-prone questions, etc., so that the most suitable learning path and learning resources such as exercises, materials, and knowledge points can be pushed to student. The system automatically adjusts the knowledge focus of the pushed learning resources according to the students' knowledge status to prevent repetitive practice of already mastered knowledge points or lack of practice of unmastered knowledge points. On the one

hand, teachers can analyze the knowledge mastery proficiency of the whole class based on the data or visual charts output by the system to create a learning status assessment report for each student and adjust the overall teaching plan in an adaptive manner. On the other hand, students can use the system to analyze their knowledge weaknesses and thus targeted appropriate exercises. Thus, adaptive learning is one of the potentially feasible solutions to the problem of "automatic assessment of students" knowledge mastery status and instructional program generation" in online education.

The purpose of this paper is to propose a knowledge-tracing-based model for recommending high school mathematics exercises, using the subject of high school mathematics as the main research context. In the subject of high school mathematics, practice exercises are the main means for students to improve their learning ability. However, in the current high school mathematics teaching, teachers or students need to find suitable exercises to practice from a huge library of exercises, which are often too large, highly repetitive and confusingly organized. There are quite a lot of low-quality unlabeled exercises in the exercise bank, which need to be manually labeled with knowledge points. Some students study through excessive exercises tactic, but this is less efficient and often results in repetition of familiar knowledge and avoidance of unfamiliar knowledge. In order to improve the effectiveness of the exercises performed by students and thus enhance the proficiency and comprehensiveness of knowledge acquisition, experienced teaching staff is needed to conduct analysis of students' knowledge status and to select appropriate exercises from the exercise bank for recommendation. The method is uneconomical and inefficient because of its high manual workload, its reliance on expert a priori knowledge, and its inclusion of a large amount of repetitive work. In addition, the traditional exercise recommendation takes the student group as the minimum granularity, but does not recommend for the knowledge mastery proficiency of specific students, which ignores the problem that different students have different learning abilities, so the recommendation is less fine-grained and ineffective for most students. The method is uneconomical and inefficient because of its high manual workload, its reliance on expert a priori knowledge, and its inclusion of a large amount of repetitive work. In addition, the traditional exercise recommendation takes the student group as the minimum granularity, but does not recommend for the knowledge mastery proficiency of specific students, which ignores the problem that different students have different learning abilities, so the recommendation is less fine-grained and ineffective for most students. In order to improve the problems of traditional exercise recommendation methods, knowledge tracing techniques can be applied to track students' learning and thus target automated exercise recommendations. The goal of this thesis is to design an exercise recommendation system based on knowledge point

annotation, knowledge tracing and resource recommendation techniques, and thus introduce an intelligent adaptive learning solution in terms of exercise recommendation.
The proposed recommendation system for high school mathematics exercises consists of three modules, which are the exercise knowledge point annotation module, the knowledge tracing module and the recommendation module. The function of the exercise knowledge point labeling module is to perform knowledge point labeling for exercises without knowledge point labeling, thus replacing the traditional manual knowledge point labeling with an automated form. The knowledge point labeling is the pre-work of the exercise recommendation, and the knowledge labeled exercises can be used as the data source of the exercise recommendation system. The knowledge tracing module calculates students' knowledge proficiency state vector, which is a representation of students' mastery of subject knowledge points, concepts and skills, by tracking students' exercise records. knowledge tracing is the core part of the system. In the final exercise recommendation module, there are two stages of matching and sorting, the former is applied to the exercise database to apply a variety of matching strategies to quickly filter the exercises and generate a collection of recommended candidate exercises, and the latter inputs the collection into the knowledge tracing system in the sorting stage for refined recommendation sorting to generate the final recommendation results.

- Chapter 2 proposes a multi-knowledge point annotation method for high school mathematics exercises based on bidirectional LSTM and graph neural network. The exercise knowledge point annotation module contains two sub-modules: exercise text mining and multi-knowledge point label classification. Since most of the exercises in the exercise database contain only unstructured data such as textual information, this paper focuses on knowledge point extraction by means of exercise text mining. It applies a bidirectional LSTM network with attention mechanism to perform exercise text mining. The exercises are firstly pre-processed by word separation, cleaning, regularization and other pre-processing steps to obtain word sequences while filtering out a large amount of interference of irrelevant information. Next, the word2vec approach is used as the learning word embedding vector instead of the simple one-hot encoding approach, which can prevent the dimensional disaster caused by the sparse input word vector matrix. Moreover, the hidden dependencies between word vectors can also be characterized by embedding learning, which is conducive to the construction of inter-knowledge point dependencies. After that, text information extraction by a bidirectional LSTM model can better solve the problem of long-range dependency of contextual elements in text. In addition, to capture inter-knowledge point dependencies on the classification model, a multi-label knowledge point labeling model based on graph convolutional network (GCN) is proposed in this paper, where each label is represented by a graph embedding of knowledge points, and the label graph is mapped into a set of intrinsically dependent knowledge point classifiers after several rounds of iterative learning. Subsequently, the word vector of the exercise text extracted from the previous sub-network is fed into the set of knowledge point classifiers to derive a multi-knowledge point prediction probability vector, thus realizing the multi-knowledge point labeling labeling task. In the experimental phase, the proposed method in this thesis is compared with a series of baseline models by conducting experiments on a self-made high school mathematics exercise dataset, and a series of multi-label classification metrics are used to compare and evaluate the model performance. The experimental results show that the method achieves more superior performance on the more complex sets of exercises with more complex knowledge point relationships.s method has achieved superior performance on the problem sets with more complex knowledge point relationships.

- Chapter 3 proposes a knowledge tracing model based on graph attention network (GAT) and Transformer architecture. The model learns the complex correlations between exercises at the knowledge point level through graph attention networks, and optimizes the traditional knowledge tracing model for the shortcoming of insufficient characterization of complex correlations between knowledge points between exercises. It solves the following problems, (1) The traditional model models knowledge points as mutually independent relationships or simplified probabilistic dependencies, while ignoring the complex graph-like relationships among knowledge points, thus performing poorly for data with complex knowledge point dependencies. (2) The traditional model cannot output the students' knowledge state, but only the probability of correct answer for the next exercise, which makes it difficult to combine with the recommendation model to recommend exercises. The proposed model combines the powerful representational learning capability of graph attention networks for data and graphs in non-Euclidean space and the remote-dependent attention mechanism modeling capability of Transformer model for serialized exercise data, which has better performance in handling longer exercise records and complex exercise datasets with knowledge relationships. In the experimental phase, the performance of the model proposed in this thesis is compared with the baseline model by conducting experiments on several publicly available datasets in the knowledge tracing domain. The experimental results show that the present model achieves better or comparable results in terms of evaluation parameters relative to most models on the publicly available datasets.

- Chapter 4 presents a mathematical exercise recommendation model based on 2 phases of Matching-Ranking. The first stage is the matching model, which is a hybrid model based on multiple matching strategies with two processes: multiplex-matching and merging. In the process of multiplex-matching, multiple matching strategies such as collaborative filtering, popularity, user preferences are used to generate the several subset of exercise recommendation candidates separately. Then, in the process of merging, these candidate subsets are merged by weighted ranking to form a final set of exercise recommendation candidates. The second stage is a knowledge-tracing-based recommendation item ranking model, in which each exercise in the set of exercise candidates obtained in the previous stage is input to the knowledge-tracing model proposed in the previous chapter for correctness prediction, and the one with the most error-prone exercises is used as the recommendation item with the highest priority. The model mainly solves the problem that traditional models often require user-initiated ratings for recommendation model construction, and cannot make efficient recommendations based on the user's knowledge mastery proficiency state. After the performance test on the public dataset and the control experiment with the baseline model, the proposed model has a better performance in tracking the students' weak knowledge mastery and can recommend the appropriate exercises based on the mastery proficiency status obtained from the tracking.

This paper analyzes the requirements of the system, rationalizes the entire recommendation system into multiple modules, and designs different neural network models and algorithms for each module to achieve and optimize the above three modules. It has both algorithm design and experimental verification methods. A certain degree of innovation. After experimental verification, it has better performance than similar models.

Keywords: Graph Neural Network, Knowledge Point Labeling, knowledge tracing, Recommended Exercises

摘 要

2010 年后，人工智能技术逐渐成为计算机技术领域的研究热点。尤其是机器围棋手 AlphaGo 的问世，引发了业界对于人工智能前景的极大关注。这带来行业的爆发式发展，也提出了大量的研究课题。在人工智能相关研究中，各种算法创新、理论突破和模型应用层出不穷，为各个行业的智能化奠定了基础。人工智能技术在教育领域的应用也催生了智能教育概念的出现。其中，自适应学习是智能教育中的热门的应用领域之一 [?]。自适应学习模型一般是通过结合对海量学生群体学习数据的大数据分析和对目标学生个体数据的精准化数据分析来追踪学生的学习状态，从而针对学生的个体特征和知识掌握熟练度来生成个性化学习路径 [?]。自适应学习技术可以将以往需要大量人工劳动的学生评估和教学计划等工作，通过自动化机器学习算法来完成，这可以系统性缓解目前国内教育资源稀缺和分配不均的问题也可以减轻教育从业者和学生的负担。它也具有极大的的发展前景和商业价值，市面上也有越来越多的人工智能研究团队和智能化教育技术公司专注于自适应学习工具的研发和应用，部分智能教育科技公司已开始将自适应学习用作其产品要核心功能或主要卖点。自适应教育技术可以综合分析学生个体层面的学习能力、知识熟练度和群体层面的热门学习资源、易错题等，从而可以将最适合的学习路径和学习资源例如习题、资料、知识点推送给学生。系统会根据学生的知识状态自动调整推送学习资源的知识侧重点，防止重复练习已经掌握的知识点或者缺乏练习未掌握的知识点。一方面，教师可以根据系统输出的数据或可视化图表来制作每个学生的学习状态评估报告分析整个班级的知识掌握熟练度，适应性地调整总体教学计划。另一方面，学生可以通过系统来分析自己的知识薄弱项，从而针对性的进行习题训练。因此，适应性学习是在线教育中 “学生知识掌握状态自动评估和教学方案生成” 问题潜在可行解决方案之一。

本文以高中数学学科为主要研究背景，目的是提出一种基于知识追踪的高中数学习题推荐模型。在高中数学学科中，练习习题为学生主要的学习能力提高手段。但是目前高中数学教学中，教师或学生需要从庞大的习题库中去寻找合适的习题进行练习，它们往往存在过于庞大、重复度高和组织混乱等问题。在习题库中存在相当多低质量的未标注知识点的习题，需要人工进行知识点标注。有部分学生通过题海战术来进行学习，但这样效率较低，且往往出现熟悉知识点的重复练习和不熟悉知

识点的缺乏练习等情况。为了提高学生进行习题练习的效果，从而提升知识掌握的熟练度和全面性，需要经验丰富的教学人员进行学生知识状态分析，从习题库中筛选出合适的习题进行推荐。该方法人工工作量大，依赖专家先验知识，且包含大量的重复性工作，因此存在不经济且低效的问题。此外，传统习题推荐以学生群体为单位，没有针对特定学生的知识掌握情况进行推荐，也没有考虑不同学生的学习能力不同的问题，因此导致推荐的效果精细度较差。为了改善传统习题推荐方法存在的问题，可以通过应用知识追踪技术来追踪学生的学习情况，从而针对性地进行自动化习题推荐。本文的目标在于设计一个基于知识点标注、知识追踪和资源推荐技术的习题推荐系统，从而推出一个在习题推荐方面的智能自适应学习的解决方案。

本文提出的高中数学习题推荐系统包括三个模块，分别为习题知识点标注模块、知识追踪模块和推荐模块。习题知识点标签模块的作用是为未标注知识点的习题进行知识点标注，从而将传统的人工知识点标注以自动化的形式代替。知识点标注是习题推荐的前置工作，经过知识标注的习题可以作为习题推荐系统的数据源。知识追踪模块通过追踪学生的习题练习记录，计算学生的知识熟练度状态向量，它是学生对于学科知识点、概念和技能的掌握度的表征。知识追踪是整个系统的核心部分。在最后的习题推荐模块，具有召回和排序两个阶段，前者应用于习题库上应用多种召回策略对习题进行快速筛选，生成推荐候选习题集合，后者输入该集合在排序阶段输入知识追踪系统中进行精细化推荐排序，生成最终的推荐结果。

- 第二章提出了一种基于双向 LSTM 与图神经网络的高中数学习题多知识点标注方法，习题知识点标注模块包含习题文本挖掘和多知识点标签分类两个子模块。由于习题库的大多数习题只包含文本信息等非结构化数据，因此本文主要通过习题文本挖掘的方式来进行知识点提取。它应用了加入注意力机制的双向 LSTM 网络来进行习题文本挖掘，习题首先经过分词、清洗、正则化等预处理步骤，得到词序列的同时过滤掉大量的无关信息的干扰。接下来，通过 word2vec 的方式而非简单的 one-hot 编码方式来作为学习词嵌入向量，这样做可以防止输入词向量矩阵稀疏带来的维数灾难等问题。而且，通过嵌入学习的方式，也可以表征词向量间的隐藏依赖关系，有利于构建知识点间依赖关系。之后，通过双向 LSTM 模型进行文本信息抽取，能够更好地解决文本中上下文元素长程依赖的问题。另外，为了在分类模型上捕捉知识点间依赖关系，本文提出了一个基于图卷积网络（GCN）的多标签知识点标注模型，每个标签由知识点的图嵌入表示，经过多轮迭代学习，将标签图映射为一组内在依赖的知识点分类器。随后，将前一个子网络提取的习题文本词向量输入知识点分类器组，得出多知识点预测概率向量，从而实现多知识点标签标注任务。实验阶段，通过在自制的高中数学习题数据集上进行实验，将本论文提出的方法与一系列基准模型进行对比，并采用一系列多标签分类指标来进行模型性能比较和

评估。实验结果显示该方法在知识点关系较为复杂的习题集上取得了更加优越的性能。

- 第三章提出了基于图注意力网络（GAT）和 Transformer 架构的知识追踪模型。该模型通过图注意力网络来学习习题间在知识点层面上的复杂关联关系，针对传统的知识追踪模型对于习题间知识点复杂关联关系表征不足的缺陷进行了优化。它解决了如下问题，（1）传统模型将知识点建模为相互独立的关系或者简化的的概率依赖关系，而忽略了知识点间复杂的图状关系，从而对于知识点依赖关系复杂的数据表现不佳。（2）传统模型无法输出学生的知识状态，而只能输出对于下一道习题的回答正确概率，从而难以结合推荐模型进行习题推荐。本文提出的模型结合图注意力网络对于非欧式空间的数据和图的强大表征学习能离和 Transformer 模型对于序列化习题练习数据的远程依赖的注意力机制建模能力，在处理较长的习题练习记录和知识关系复杂的习题数据集上具有更佳的性能。实验阶段，通过在若干个知识追踪领域公开数据集上进行实验，将本论文提出的模型与基准模型进行性能对比。实验结果显示，在公开数据集上，本模型相对于大多数模型在评估参数上都取得了较优或者相当的结果。

- 第四章提出了基于召回-排序两阶段的数学习题推荐模型。第一阶段为召回模型，它是一个基于多召回策略的混合模型，它具有多路召回和融合两个过程。在多路召回过程，采用了基于协同过滤、热门度、用户偏好等多个召回策略用于分别生成习题推荐候选集合。然后在融合过程，将这些候选集合进行加权排序合并，形成一个最终的习题推荐候选集合。第二个阶段为基于知识追踪的推荐项排序模型，将前一阶段获取的习题候选集合中的每个习题输入到前一章提出的知识追踪模型，进行正确率预测，将最容易出错的习题的作为优先级最高的推荐项。该模型主要解决的是传统模型往往需要用户主动评分来进行推荐模型构建，而无法基于用户的知识掌握熟练度状态进行高效推荐的问题。经过在公开数据集上的性能测试和与 baseline 模型的对照实验，提出的模型在对于学生的掌握薄弱知识的追踪性能较为优越，并能依据追踪得到的的知识掌握熟练度状态推荐合适的习题。

本文通过分析系统的需求，将整个推荐系统合理化分为多个模块，并针对各个模块设计了不同的神经网络模型和算法来实现和优化上述三个模块，在算法设计和实验验证方法方面都具有一定的创新性。经过实验验证，具备相对于同类模型更好的性能。

关键词： 图神经网络，知识点标注，知识追踪，习题推荐

Acknowledgments

After two years of study at the master's level, on the occasion of completing my dissertation, I sincerely express my sincere gratitude to my teachers, faculty leaders, counselors, classmates and friends who have helped me.

First of all, I would like to give special thanks to my supervisor, Prof. Zhi-Feng Wang, who has given me patient and meticulous guidance and great help in the process of completing my usual scientific tasks. He also provided quite a lot of useful and enlightening suggestions during the selection of the topic, conception, experimental simulation and thesis writing of this paper. Finally, my thesis has been improved continuously. This is all thanks to Mr. Wang's diligent guidance and teachings.

Then, I would like to thank all the leaders and teachers of the college for their considerable and valuable advice during my work and study at the master's level, which has refined my heart and for which I am full of gratitude. This is certainly an invaluable asset for my future career.

I would also like to thank all my classmates and friends who have walked with me through my research study career in the past three years, and we have helped each other in our study, motivated and cared for each other. Thank you for their friendship and support!

Publications

List your publications here...
Please remove this page if you do not have any...

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background and Significance

The development of technological research and commercial application deployment of artificial intelligence has shown an accelerating trend in recent years. The deployment of various algorithms related to AI and big data analytics based on artificial intelligence plays a positive role in accelerating the digitization of enterprises and organizations, improving the structure of industry chains and increasing the efficiency of information utilization. As a traditional service industry, the education industry also has considerable room for intelligent revolution. In traditional education models, student groups are often divided into basic education units like classes. Therefore, in a class, the granularity of all teaching activities is equivalent to the size of the class, which leads to the fact that the content of students' learning does not completely match their needs. That is, the knowledge points that the students have mastered are over-practice or the knowledge points that the students have not mastered lack practice. This has caused students to be tired of learning, anxiety and other conditions, which have a greater impact on their learning efficiency and learning effect. On the other hand, for teachers, generalized detailed monitoring and evaluation of individual knowledge status requires a huge workload, so teachers often only pay attention to a part of the students, leading to the neglect of the learning situation of most students. This has a greater impact on students' learning enthusiasm and learning conditions.

In recent years, China has released a series of policies to promote the application of artificial intelligence in education. AI technology for student learning monitoring and tracking can free up teachers' labor and also solve problems that are difficult to solve in traditional manual education, thus greatly improving the quality and efficiency of education and realizing smart education. In smart education, adaptive learning is a model that has been proven in practice, and it already has many successful cases of commercial deployment. A considerable number of online education platforms have deployed adaptive learning education system services in different domains. It uses technologies such as data analytics, machine learning and automation to automatically assess and track students' knowledge status by analyzing their learning behavior data. In addition, the system provides students with learning path planning and learning resource recommendation services by combining personalized information such as students' potential and areas of expertise. It can improve teachers' teaching quality while reducing teachers' work pressure, and can also effectively help students improve their learning efficiency. The exercise recommendation system is an implementation of an adaptive learning model, which includes two parts: learner knowledge mastery proficiency modeling and exercise recommendation. The general model of knowledge mastery modeling is to input the semantic records of learners' learning interactions such as question records, quiz records, etc. to capture the learning characteristics of learners and achieve the dynamic tracking of their knowledge mastery proficiency. The exercise recommendation part analyzes the learners' knowledge mastery proficiency model and recommends exercises that are most relevant to the learners' relatively weak knowledge mastery.

High school mathematics is a relatively difficult subject at the high school level. At present, in high school mathematics, the knowledge points are complicated and closely interconnected, and the corresponding library of exercises is large and confusing. As a result, many students do not know how to reasonably assess their knowledge and practice in a targeted manner, leading to the emergence of "excessive practicing tactic". This has a negative impact on students' interest in learning and self-confidence. The purpose of this paper is to propose a system for tracking students' knowledge of mathematics in real time and recommending exercises according to their knowledge status. The system is divided into three parts, the first part is the knowledge labeling of the exercises as the data preparation part of the exercise recommendation system, the second part is the core knowledge tracing model, which tracks the knowledge proficiency status by inputting students' records of doing exercises, and the third part is the functional part of the system, which recommends exercises targeted by students' knowledge proficiency output from the knowledge tracing model. The system can effectively achieve the goal of adaptive learning

1.2 Research Status

The first part of this paper is a graph neural network and natural language processing (NLP) based knowledge point labeling for high school math exercises, the second part is a graph attention network and Transformer based knowledge tracing model, and the third part is a matching and ranking based exercise recommendation module. Thus, the techniques covered in this paper include high school mathematics subjects, graph neural networks, natural language processing, multi-label classification, recommendation systems, and other research topics. Next, this section reviews the current state of research on these techniques.

1.2.1 High school Mathematics

Mathematics is an essential subject in both basic education and scientific research. Mathematics is a science devoted to the study of relationships between quantities and spatial forms, with a more complete system of symbols, a clear and unique structure of formulas, and more vivid and intuitive verbal expressions such as words and images. In mathematics, the establishment of knowledge structure and cognitive structure plays a considerable role in the learning of the subject. The "cognitive structure" denotes the organization of declarative knowledge among the human brain, while the cognitive structure internalized through learning and displayed through network structures or graphics is the knowledge structure [**?**]. Most of the knowledge contents that learners need to learn come from the experience summaries of previous people in practical activities, and the process of their learning is the cognitive learning of these summarized knowledge, and constantly digesting, adjusting and reorganizing the structure of knowledge, so as to build a more perfect and suitable knowledge structure, which is also a process of combining with innovative thinking. In mathematics, the mastery of knowledge often comes from practice, such as exercises, proofs and derivations, and so on. As a major subject at the secondary level, mathematics is characterized by a high degree of abstraction, rigorous logic and extensive applications. The body of knowledge in this subject is built up by using many abstract concepts, and new abstract concepts are formed by learning and expanding on these concepts. In addition, mathematics is very logical because any conclusion reached in mathematics requires rigorous logical reasoning and strict proof before it can be considered reasonable. It is an important tool for social practice or scientific research, and the study of mathematics is indispensable in all walks of life and in all areas of society. Mathematical knowledge is also intrinsically related to each other, so that they can be arranged and learned in a specific logical progression in the specific learning process. These intrinsic relationships can be classified as synonymous, antecedent, successor, inclusion, brotherhood, opposition, etc.[**?**] By analyzing the knowledge points, a network of knowledge point associations can be established to facilitate subsequent knowledge mastery status tracking.

1.2.2 Graph Neural Network

In recent years, the rapid development of neural network models has driven research related to machine learning, and various neural network paradigms have been designed for various application environments and tasks, such as convolutional neural networks (CNNs), which are widely used in pattern and feature recognition, and recurrent neural networks (RNNs), which are applied to serialized data learning. Traditional neural networks have good computational and processing capabilities for data in Euclidean space such as language, sequences, and images, but have limitations for non-Euclidean space such as knowledge networks. In this study, the object of study is mathematical subject knowledge, and the knowledge points form a net-like correlation between them. Therefore, the complex relationship between knowledge points cannot be fully characterized by traditional neural network model processing, and it is more compatible with the natural paradigm to learn the knowledge point network by graph. Each knowledge point or exercise can be treated as a node of a graph, and the edges between nodes represent the association between knowledge points or exercises. Therefore, this paper introduces graph neural network to capture the correlation between knowledge points and exercises, which can reasonably abstract the data in a higher dimension and achieve better results.

At present, for the graph structure, the machine learning tasks that can be performed can be roughly divided into the following types:

1. Graph node classification task: For a graph in which each node has a corresponding feature, and the categories of some nodes are known, a classification task can be designed to classify unknown nodes.

2. Graph edge structure prediction task: For a graph where the edge relationship between some nodes is known, the edge structure and relationship of the location are mined based on the existing information. This type of task is the edge prediction task, that is, Prediction of the relationship between nodes and nodes.

3. Graph classification: For the entire graph, we can also classify the graph. Graph classification is also called the graph isomorphism problem. This is often achieved by aggregating the node characteristics of the graph and then classifying it.

The graph neural network was proposed in 2009 by Franco et al. The model is based on the theory of immobile points and aims to obtain the hidden state of each node. However, when the number of stacks is too many, the state convergence tends to be too smooth and makes it difficult to learn the feature information of the graph. Graph neural networks are generally used in an iterative manner to compute data transfer to nodes as a way to learn the target node representation. The concept and application of graph neural networks have undergone continuous development, and new graph neural network models have been proposed one after another. In recent years, the increase in computing power of parallel computing devices such as GPUs has made it possible to apply many graph neural network models that were too limited by computing power. In this paper, graph convolutional neural networks (GCN) and graph attentional neural networks (GAT) are mainly used, etc. GCN is a model proposed by Kipf in 2016 for semi-supervised classification [**?**], and the computation of GCN is based on a hierarchical structure, where each layer is the result of feature extraction from the previous layer, from the node level, and the nodes propagate each other hiding the state. The final GCN output undergoes multiple layers of abstraction, and in terms of learning of node feature information and graph structure information, GCN achieves State of the Art (SOTA) performance on almost all of the datasets related to most public node classification or edge prediction. GAT was proposed by Veličković et al. in 2018 [**?**], the network introduces a self-attentive mechanism in the propagation process, where the hidden state of each node is computed by paying attention to its neighboring nodes. The network uses a local network design structure, so that only neighboring nodes are computed during the computation, reducing the computational load.

1.2.3 Knowledge tracing algorithms

Knowledge tracing (KT) is a technology that models student knowledge acquisition based on past answer records and results. It is a typical model for modeling learner knowledge acquisition, and has evolved into the mainstream approach for modeling learner knowledge acquisition in intelligent tutoring systems. The main task of knowledge tracing is to analyze learners' knowledge mastery based on their historical learning records and thus automatically track changes in students' knowledge levels over time in order to be able to accurately predict students' performance in future learning and provide appropriate learning tutoring. In this process, the knowledge space is used to describe the level of student knowledge acquisition. In the process of knowledge tracing, the knowledge space is modeled as a collection of concepts, and students' mastery of a portion of the collection of concepts constitutes students' mastery of knowledge. Some educational researchers argue that students' mastery of a specific set of related knowledge points affects their performance on exercises, i.e., the set of knowledge students have mastered is closely related to their external performance on exercises. Teachers can assess students' knowledge status to better understand where their mastery is weak and target their instructional programs.s

In the knowledge tracking model, the traditional approach is implemented by cognitive diagnosis, which is an algorithm for modeling students' proficiency in knowledge points.

Among them, Item response theory (IRT)[**?**] models learners based on a one-dimensional continuous model, while the DINA model models students' knowledge states based on a series of vectors that represent users' knowledge mastery related to the exercises[**?**]. Besides, Bayesian knowledge tracking (BKT) is a more widely used knowledge tracking model based on probabilistic graphical modeling [**?**], which models the learner's knowledge state as a set of binary variables representing whether the knowledge points are mastered or not, and the algorithm uses Hidden Markov Model (HMM) to maintain the knowledge proficiency variables. However, the drawback of BKT is that it ignores the forgetting property of students for knowledge and the correlation between knowledge points. In 2015, Deep Knowledge Tracking (DKT) was proposed[**?**], which is the first algorithm to apply RNN models to the knowledge tracking task, and the model uses LSTM to track the dynamic change of students' knowledge proficiency over time and learn the student's knowledge mastery vector, which is used to predict students' performance on questions. And the Dynamic Key-Value Memory Networks for Knowledge Tracing (DKVMN) model proposed in 2017 borrows the idea of memory-enhanced networks, which can store students' mastery levels of knowledge concepts using a key matrix by using the relationships between the underlying concepts, so the model can Explicitly output the student's mastery of the knowledge point. The model refines the relationship between exercises and knowledge points, and achieves better performance than DKT and BKT on public datasets. However, these models do not appropriately model the complex correlation of knowledge points, and thus degrade the prediction performance for exercises with complex knowledge point associations. In contrast, graph neural network, as a model adapted to model non-Euclidean spaces, can be a solution to model the relational network of knowledge points. The Graph-based Knowledge tracing model proposed at ICLR2019[**?**], which uses graph nodes to model student's responses to the exercises, considers the effect of doing questions on the knowledge state as well as similar exercises, and the test results in the public dataset tabulate the performance of this model over DKVMN.

1.2.4 Recommendation System

In the early days of the Internet, users often searched for content of interest. However, with the massive growth of Internet information, it has become more and more difficult to search for suitable content. Therefore, it is based on pushing users the information they are interested in. Recommendation system technology was developed. At present, the recommendation system has been widely used on the Internet, and it has brought huge benefits to both service providers and users. In the field of education, for the majority of students, the existing exercise database is too large, so students often experience information tragedy in the process of extracurricular exercises. The learning efficiency of adopting the sea tactics is low and the results are slow. Therefore, a system for adaptive exercise recommendation based on students' knowledge status is necessary and important[?].

There are several types of algorithms in recommendation systems: collaborative filtering, content-based recommendation models, etc. The core idea of collaborative filtering algorithm is to divide users into groups based on similar interests, and according to this principle to find users with similar interests to the recommended person, and recommend the content that the group is interested in to the user. The collaborative filtering algorithm integrates similar users' evaluation of recommended items to form a prediction model of users' interest in the item. Collaborative filtering is one of the most widely used and successful recommendation algorithms. Collaborative filtering can be further subdivided into two types: user-based model and item-based model[?]. Among them, the former focuses on finding similar users, i.e., recommendation subjects, and then recommending them through the content of interest to similar recommendation subjects, while the latter focuses on finding similar items, i.e., recommendation items, and filtering out the most similar recommendation items as recommendation results by calculating the item similarity. In collaborative filtering, the calculation of similarity is a key issue. There are already similarity algorithms such as Jaccard coefficient[?], cosine similarity[?], related similarity[?], etc. Collaborative filtering often has "cold boot" problem, i.e., when there are fewer similar users or entries, a large degradation of performance occurs. Therefore, random recommendations or other content recommendation methods can be used in the starting phase of the recommendation model. The content-based recommendation model, on the other hand, performs interest pattern mining based on the characteristics of the user's interest entries, which essentially builds a model for calculating the degree of interest in an item, and the method does not depend on other users, but the model is prone to the problem of duplicate recommendations, and thus requires an additional step for de-duplication operations[?]. macos/deepLFree.translatedWithDeepL.text

1.3 Research Objectives and Content

The purpose of this study is to propose a high school exercise recommendation system based on knowledge tracking, which is divided into three parts, the first part is the exercise knowledge point labeling part, the purpose of this part is to classify knowledge point labels for many unlabeled exercises, the model is a multi-knowledge point labeling classification problem, in this section need to text mining for Chinese mathematics exercises, establish the knowledge point labeling model, and the model In this section, we need to perform text mining on Chinese mathematical exercises, build a knowledge point labeling model, and test the model performance. In the second part, the labeled knowledge point exercises are used as the input of the knowledge tracking model, which learns the knowledge graph embedding representation of the exercises through graphical neural networks, designs a knowledge tracking model based on the Trasnformer structure to track the students' knowledge mastery proficiency state and predict the correct probability of the next exercise, and outputs the hidden knowledge state vector, which is used in the later recommendation section as the next level input. In the last part, a recommendation model based on two stages of matching-ranking is designed to first filter the exercises to filter the candidate recommended exercises, then input the knowledge tracking model to predict the probability of correct answers, and output a final list of recommended exercises according to the ranking rule that the higher the error rate is, the higher the priority is.

1.4 Thesis Organization and Structure

Chapter 1 of this thesis is an introduction. The research background of the study, the research progress and review of related techniques and theories are presented. The research focus and objectives of this paper are described. Based on the currently available models for requirements analysis, a solution based on knowledge tracking exercise recommendation system is proposed and divided into three modules: exercise knowledge point annotation, knowledge tracking and exercise recommendation.

Chapter 2 of this thesis focuses on the labeling of exercise knowledge points. In the exercise resource recommendation system, the knowledge points of the exercises need to be parsed, and then based on the students' current knowledge mastery, it is recommended that the students have insufficient knowledge points and related exercises. The algorithm model in this chapter is divided into two parts: text information extraction of exercises and labeling. In the experimental part, the effectiveness of the model is verified by comparing with several traditional models.

Chapter 3 of this thesis proposes a knowledge tracing model based on graph neural network. The model is divided into three parts: exercise-knowledge point relationship embedding learning, knowledge state encoding and answer prediction decoding. The article first introduces the design ideas and related technologies theoretically, and then compares the benchmark model with the public data set in the experimental part to verify the effectiveness of the model and evaluate the performance of the model.

Chapter 4 of this thesis proposes a recommendation system model based on two phases of matching and ranking is proposed. In the matching phase of the model, multiple candidate exercise subsets are generated separately by a multiplex-matching algorithm, and then merged into the final set of candidate exercises by weighted ranking. In the ranking phase of the model, the exercises in the candidate set are used as the input to the knowledge tracking model, and then the correct probabilities of the exercises are collected and prioritized to obtain a list of recommended exercises.

Chapter 5 of this paper presents the conclusions and future directions for improvement of each part of the model.

Chapter 2

Exercise Knowledge Point Mining Based on Graph Neural Network

2.1 Research Motivation

All aspects of the recommendation system, including data collection, data mining, and data recommendation, are predicated on establishing a high-quality data source. In this thesis's study topic, the establishment of a standardized and structured exercise database is the key first step to build a test recommendation system. The construction of an exercise corpus is also a complex and challenging task, requiring comprehensive consideration of all aspects of the exercise data and adding enough additional information to the exercises for subsequent data mining. High school mathematics has hundreds of knowledge points, and on average, each point has dozens of exercises of different difficulty gradients from easy to difficult. The size of a quality math subject exercise corpus is 100,000 scale. In addition to the quantitative requirements, there are two fundamental issues in a quality exercise corpus, in addition to the quality of the questions and the matching of the educational content (textbooks), one is the construction of the knowledge point relationship network, and the other is the construction of the knowledge point-exercise relationship. Moreover, it is the optimization of these factors that leads to the construction threshold of high-quality exercise corpus and the extremely high cost. The question databases on the market often lack attention to these factors, leading to the emergence of many poor-quality exercise databases. Therefore, the labeling of the knowledge points of the questions and the construction of the knowledge system is one of the most central issues in building a quality exercise corpus.

One of the critical problems in building a test recommendation system is recommending topics in conjunction with knowledge points. When referring to the solution to this problem, the knowledge point labels should be considered. In textbooks and syllabi and various teaching aids, there are various descriptions of “knowledge points”. Knowledge points in high school mathematics, such as functions, definition domains, value domains, or analytic equations, have direct concepts, applications of methods, abstractions of topics, and summaries of similar solutions. There is no one standard way to classify these, and the methods of knowledge point system construction may be very different. As far as the task of data mining is concerned, this thesis is more concerned with the concepts and skill points it involves, and these can be concluded for confidential information through the text of the exercises. As an exercise recommendation system that analyzes students’ knowledge mastery proficiency, the knowledge point labels should be able to describe the core knowledge points, methods, or ideas of the topic quiz and be able to distinguish the ability requirement points for students to build a more robust user knowledge mastery model and recommendation engine.

In this chapter, the knowledge point labels are mined for topic recommendation and analysis reports. He requires to refine the knowledge point association of the topic, i.e., several corresponding knowledge point tags for the topic, where there will also be dependencies between the knowledge points, and also, to build a complete knowledge map of knowledge points for generating student learning reports. The first problem is a knowledge point mining task using the topic text, which is also a classification task. To be more specific, it is a hierarchical classification task based on the information in the short text of the topic. In this task, the most basic technologies of natural language processing (NLP) and machine learning should be used. By learning a large amount of manually labeled topic text and knowledge point labeling results (also called training corpus) - obtaining the features of the topic text through NLP techniques and obtaining the classification model through machine learning. The system can do knowledge point classification automatically. In this definition, the object to be classified is a topic (including stem, answer, paraphrase, etc.), and the result is a set of knowledge point labels. The input to the learning system is a set of training exercises, i.e., n questions that have been labeled with their corresponding knowledge labels; the learning system trains the given classification model based on the training data. In the prediction phase, the exercises’ input to be labeled is used to output a set of predicted labeling results for the knowledge points of the exercises.

2.2 Proposed Model

2.2.1 Algorithm Overview

This section aims to construct a model for mining the exercise-knowledge point relationship and a model for mining the association between knowledge points. Establishing an exercise-knowledge point relationship means labeling the knowledge points of an exercise, a collection of knowledge concepts used to understand and solve the exercise. Therefore, accurately describing a test question's knowledge points is essential for the subsequent knowledge tracking and recommendation process. The two basic classification approaches that already exist are expert labeling and machine learning. The former means that education experts combine their professional knowledge to annotate knowledge points of test questions. However, when the number of questions or complexity of questions is high, the manual annotation has problems such as high workload, high subjectivity, and imperfect annotation. Also, considering the association between knowledge points, manual annotation also has the problem of not taking into account the inline knowledge relationship. Another way is to use the rule-based automated annotation method, which performs knowledge keyword matching by non-intelligent means such as text pattern matching. However, many exercises often do not have explicit knowledge point texts, so the correctness rate of this method is not satisfactory. Also, an exercise often has multiple knowledge points, so in effect, knowledge point mining is a multi-label classification problem [**? ? ?**]. This chapter also discusses "how to effectively model the relationships between knowledge points" and proposes a multi-label classification model based on graph neural networks.

In 2019, a multi-label image classification model was proposed [**?**]. Inspired by this model, this chapter proposes a multi-knowledge point labeling model based on attention mechanism exercise description text feature extraction and graph convolutional neural network (GCN) based knowledge point relationship mining, which builds a knowledge point relationship graph to describe the association relationship between knowledge points by a data-driven approach, builds classifiers for knowledge points separately, and then performs multi-label classification. Its main architecture diagram is as follows Fig.**??**.

From the structure, it can be seen that there are generally two parts to the model:

1. The first part is the exercise description text information mining module, which uses a Bi-LSTM network with an added attention mechanism to text-mine hidden knowledge information on questions (including question descriptions and answers). In this thesis, an end-to-end network training approach is designed to achieve the overall iterative learning of the model. Specifically, it consists of a text preprocessing part that performs the text subdivision, filtering, and deduplication parts, an embedding layer that

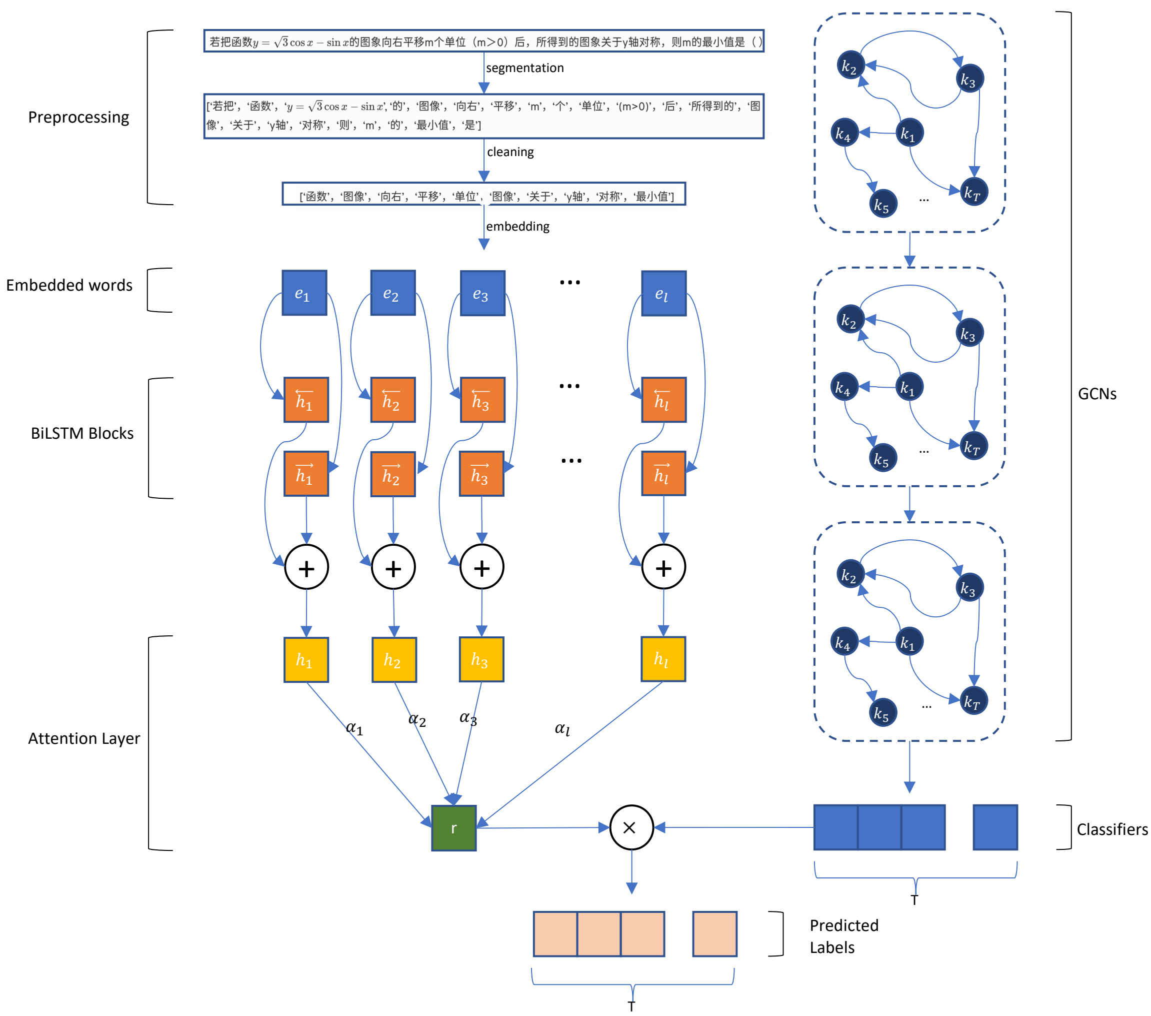


Fig. 2.1 Structure of the Knowledge Tagging Model

performs the word vector embedding calculation, and an Attention-based Bi-LSTM (Bi-LSTM) network that performs text information mining, which was proposed by Peng et al. in 2016 [**?**], outputting a textual information representation vector.

2. The second part is a GCN-based knowledge point association multi-label classifier, which maps knowledge points to a set of interdependent target classifiers. These classifiers are computed with the exercise information representation vector outputted by the first part to output a result vector representing the labeling probability of each knowledge point.

2.2.2 The Exercise Description Text Mining

This section is based on the Attention based Bi-LSTM, whose architecture design is shown in Fig.**??**.

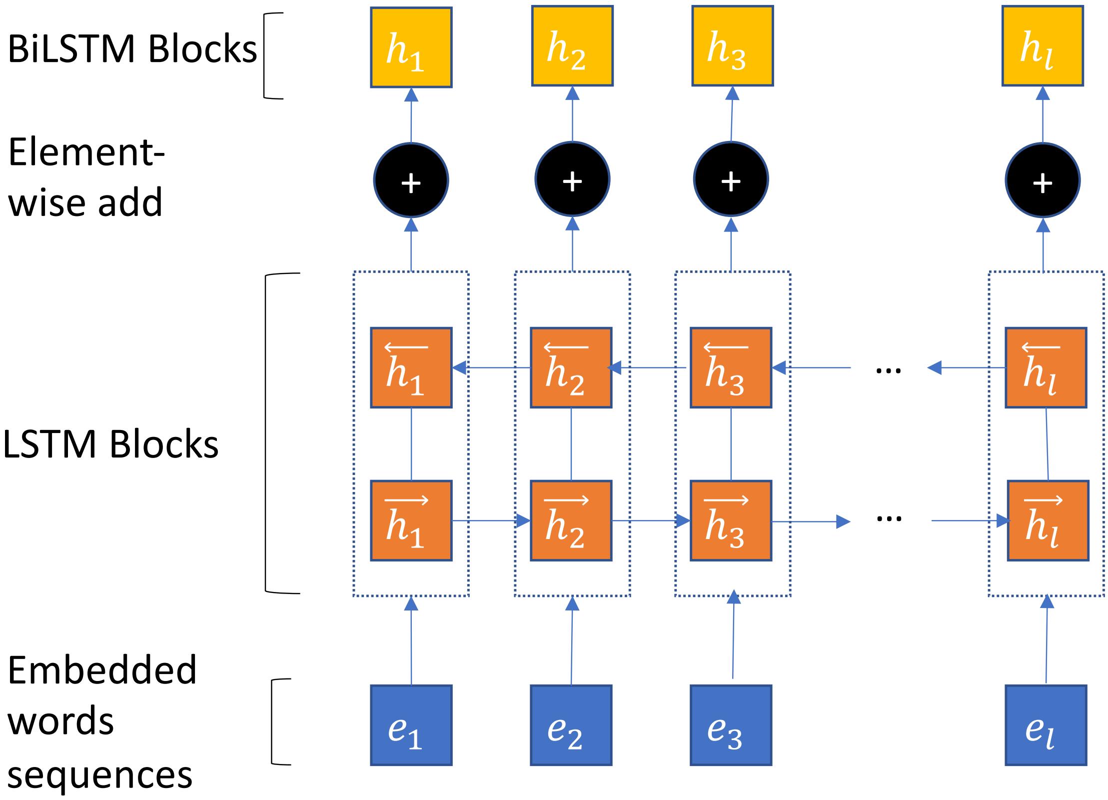


Fig. 2.2 Structure of Attention Based Bi-LSTM

The model includes four layers: Preprocess Layer, Embedding Layer, Bi-LSTM layer, Attention Layer, and Output Layer:

Pre-process Layer

The preprocessing stage mainly includes word separation, cleaning, and regularization. Considering that our research object is Chinese high school mathematics test questions, compared with English, there is no middle space in the middle of sentences of Chinese language, so it is necessary to use the word separation algorithm to decompose the sentences into subwords. There are many texts in the content that are irrelevant to the sentence expression, which will cause much interference and redundant information if the calculation is performed directly, so additional text cleaning is also a necessary step. The Fig.**??** shows an example of preprocessing of an exercise description text.

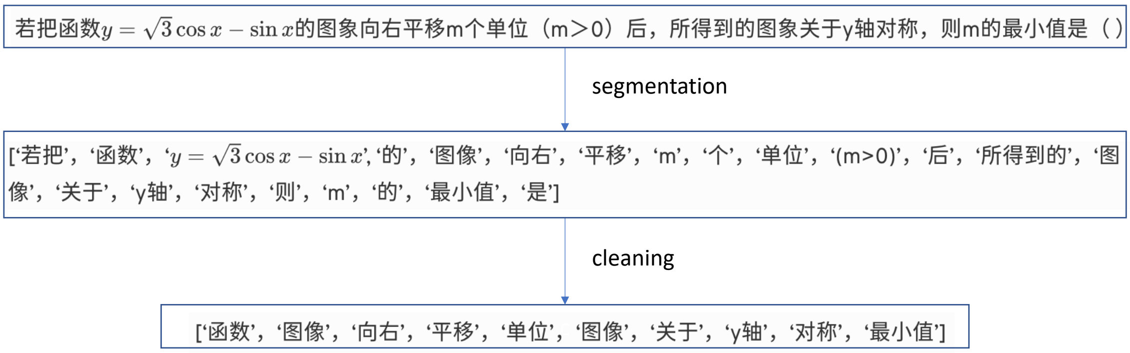


Fig. 2.3 Example of Preprocessing

- Segmentation: Compared to other phonetic languages such as English and Spanish, Chinese language preprocessing requires an extra step of word separation. There is no natural separation between words in Chinese sentences. Therefore artificial segmentation of sentences according to word meanings is required. Choosing a suitable Chinese word segmentation algorithm can lay the foundation for the subsequent processing and improve the overall model performance. At present, the Chinese word segmentation is mainly divided into two types of algorithms: rule-based lexical matching and statistical model-based. Compared with the former, the latter has better generalization ability and learning ability. It performs better for word separation beyond the a priori rules, such as ambiguous words and unregistered words. In this model, the currently popular jieba subword is used to implement a word graph scan based on the Trie book structure, and algorithms such as stage planning and HMM model are used to find the maximum subword grouping and merging based on word frequency to achieve the identification of future logged-in words. Users can also customize stop words and user dictionaries from achieving proper noun recognition.

- Cleaning: Corpus cleaning preserves useful data in the corpus and deletes noisy data. Common cleaning methods include manual deduplication, alignment, deletion, and labeling. For words that are not necessary for a sentence, i.e., stop words, their existence does not affect the sentence's meaning. There will be a large number of function words, pronouns, or verbs and nouns with no specific meaning in the text. These words are not helpful to the text analysis so that these stop words can be removed. For the exercise description text, there are many mathematical expressions, symbols, etc. Considering that these expressions in many exercises are not in text format, OCR technology must be applied to preprocess mathematical expressions from pictures to text. Therefore, when the Chinese text is sufficient, mathematical expressions can be removed to reduce the calculation load.

After the data processing step, a clean sequence of text tokens is obtained, and next in the Embedding layer, the BERT technique can be used to perform text embedding operations.

BERT-Based Embedding Layer

In applying deep learning, embedding as a preprocessing approach for generating embedded vectors brings a great extension to the application of neural networks in various aspects. In applying deep learning techniques, embedding is an instrumental skill because it reduces the spatial dimensionality of a discrete variable and allows a meaningful representation of that variable. For example, in NLP, if basic one-hot coding is used, it often results in too many dimensional and sparse vectors and also fails to learn the dependencies between vectors. During embedding training, the embedded vectors are updated, which can clearly show the exercises between the vectors. The one-hot encoding is the most naive method, which turns all words into binary patterns, i.e., all words are only present or absent in two cases, so each word is a binary vector with only 1-value and all other 0-values. When the amount of words is large, this vector's length will also be quite long, and the subsequent computation will also generate a large number of invalid computations, i.e., the sparse matrix computation problem. word2Vec was proposed by Google Language 2013 [**?**], which predicts its context by words or predicts words by contexts, a static word embedding learning model, but encounters bottlenecks in solving problems such as polysemous words. The Bidirectional Encoder Representations from Transformers (BERT) model proposed by Google in 2018 utilizes the Transformer as the base unit to pre-train masked language models, achieving State of the art (SOTA) performance in almost all NLP tasks. It has a powerful semantic representation effect. In this thesis, BERT is utilized as a word Embedding vector learning module to achieve greater generalization and adaptive capabilities in different contexts of different idiomatic texts.

In the exercise description text, there will be some combinations of words, i.e., groups of words with indivisible lexical meanings formed by combining multiple words, and since the words are masked during the training sample generation phase of BERT, these combinations may be masked separately, resulting in ambiguity and causing training performance degradation. Therefore, this thesis applies the processing of Whole Word Mask (WWM), which was proposed by Cui et al. in 2019 [**?**]. By applying WWM, when one part of a combined word is masked, then the other parts of that combined word are also masked, as shown in Fig.**??**.

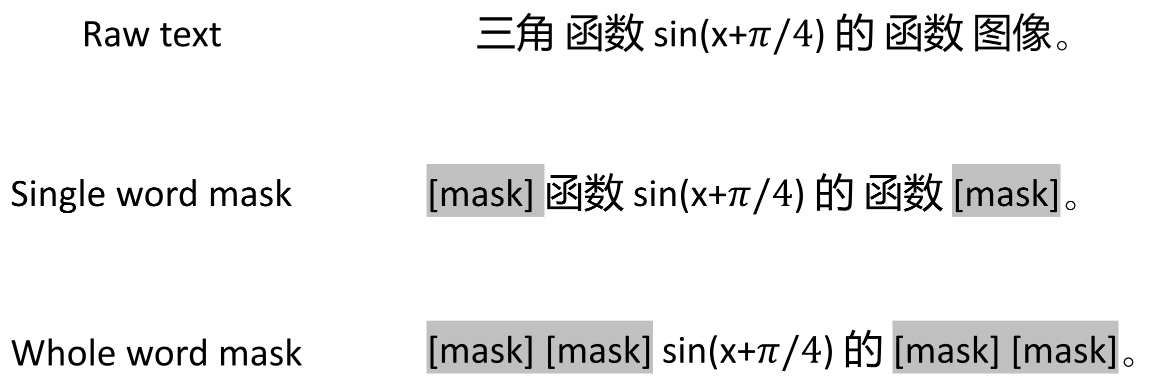


Fig. 2.4 Whole word mask

Some words of the original sequence of exercise description text words obtained after the word separation process are processed by WWM, and the marker [CLS] is added at the beginning of the sequence, and the inter-sentence is marked by [SEP] separator. After training, the word embedding vector is output. The output embedding of each word consists of three parts: token embedding, segment embedding, and position embedding, which characterize the word's embedding information from different perspectives. Subsequently, the word embedding vector is fed into the feature extraction bidirectional Transformer layer of BERT, and the feature sequence representation vector containing deep semantic features can be obtained. The overall architecture is shown in the Fig.**??**.

Bidirectional LSTM Layer

RNNs are well equipped to solve various types of problems and tasks in serialized pattern data modeling. In this section, the core of the task is a sequence-to-sequence (Seq2Seq) task, where an embedding vector described by the topic is input and a sequence containing the currently trained information is output. The most rudimentary idea is to use the original

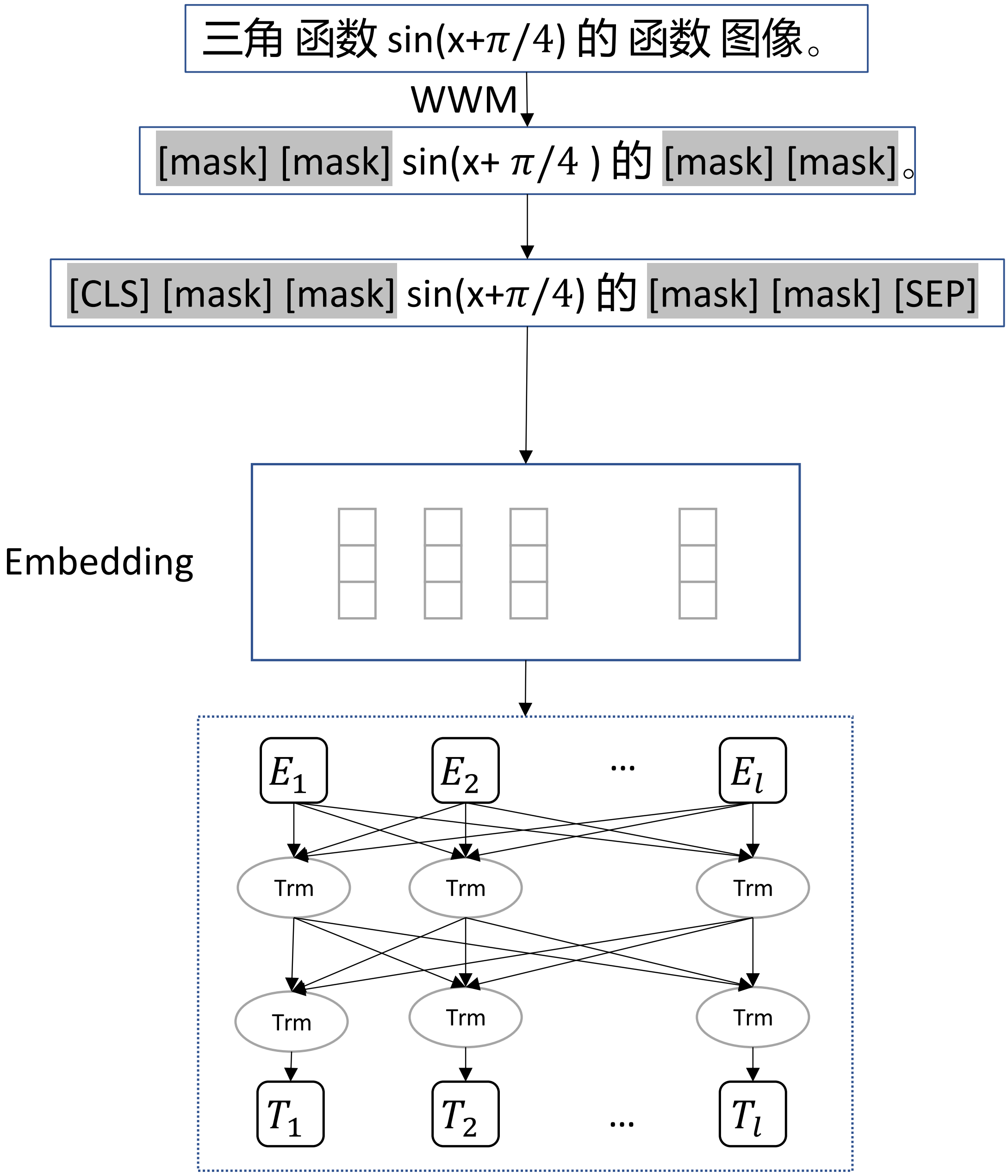


Fig. 2.5 BERT-based Embedding

RNN. Its structure is shown in the Fig.fig:ch2-rnn-model, where x_t, h_t and y_t denote the input, hidden and output values at time t, respectively. Then, the RNN training formula can be written as **??**:

$$\begin{aligned} h^{t+1} &= f(W_t^h h^t + W^i x^t) \\ y^{t+1} &= f(W^o h^{t+1}) \end{aligned} \tag{2.1}$$

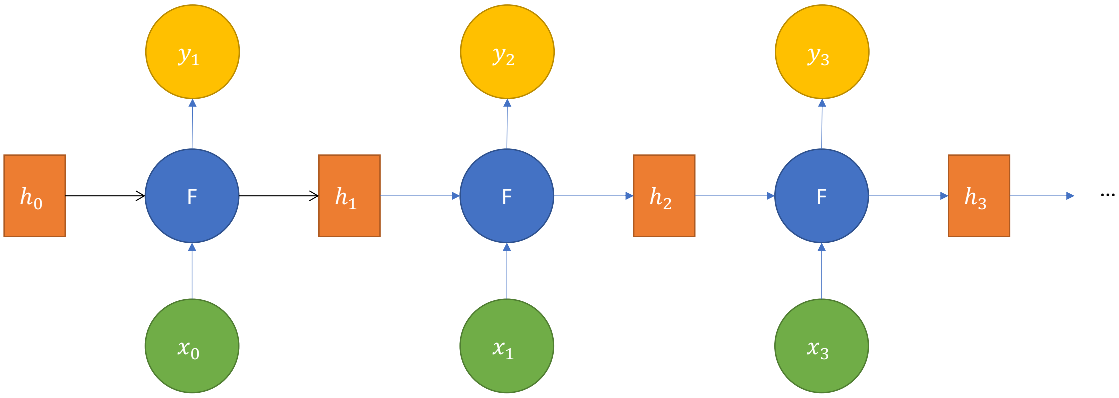


Fig. 2.6 Naive RNN Unit

However, when the sequence is long, the problem of long dependencies arises. For example, for a sequence, the current state depends on a state far away from the current state, and as the time interval increases, the ability of RNNs to learn state representation is greatly reduced. Long short-term memory (LSTM) [**?**], as a solution to overcome shortcomings of RNN, which uses a gating mechanism to achieve long-term memory. It solves the problems such as gradient explosion or gradient disappearance of RNN. It also has a good performance in capturing sequence information. The general model of LSTM is like Fig.**??**, which represents the computational details within an LSTM cell at the moment of t. Among them, σ is the activation function, c_t is the cell state representation, f_t is the forgetting gating calculation, i_t is the input gating calculation, o_t is the output gating calculation, and h_t is the hidden state representation. The forgetting mechanism controlled by gating can be effectively modeled for long-range sequential unitary information.

One-way LSTM also has an inherent drawback that it can only capture the sequence state information before t moment, i.e., it can only capture the previous sequence input. The bidirectional LSTM (Bi-LSTM) [**?**] can capture semantic information in both directions by feeding the reverse sequence into the LSTM and aggregating it with the forward LSTM sequence, while modeling the dependencies in both directions. The Bi-LSTM output can be obtained by inputting the positive sequence and the reverse sequence input sequence into two sets of LSTM networks and perform element-wise addition. The output of positive-order

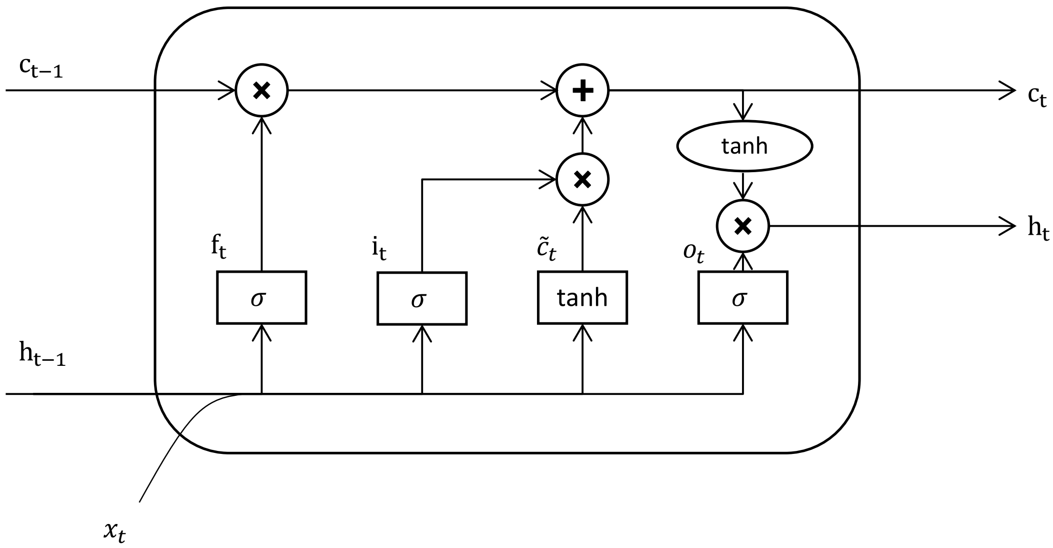


Fig. 2.7 Structure of LSTM Unit

LSTM is $\overrightarrow{h_t}$, the output of reverse-order LSTM is $\overleftarrow{h_t}$, $\bigoplus$ means sequence concatenation. The output of Bi-LSTM is:

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(e_t) \tag{2.2}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(e_t) \tag{2.3}$$

$$h_t = \overrightarrow{h_t} \bigoplus \overleftarrow{h_t} \tag{2.4}$$

The Bi-LSTM Structure is like Fig.**??**.

Here, the Bi-LSTM output a bidirectional sequence as the concatenation of positive-order and reverse-order LSTM output sequence.

Attention Layer

Since each feature word's impact on the overall semantics in a text mining task is asymmetric, i.e., keywords have a decisive role in determining the meaning of the entire text. The proposed Attention model proposed by Bahdanau et al. is based on [**?**]. Human attention is a mechanism that focuses on key information ignoring non-key information, i.e., individual information points are weighted differently. This method achieved remarkable results in different tasks such as image vision [**? ?**], language mining [**?**], and voice recognition [**?**] and is applied widely.

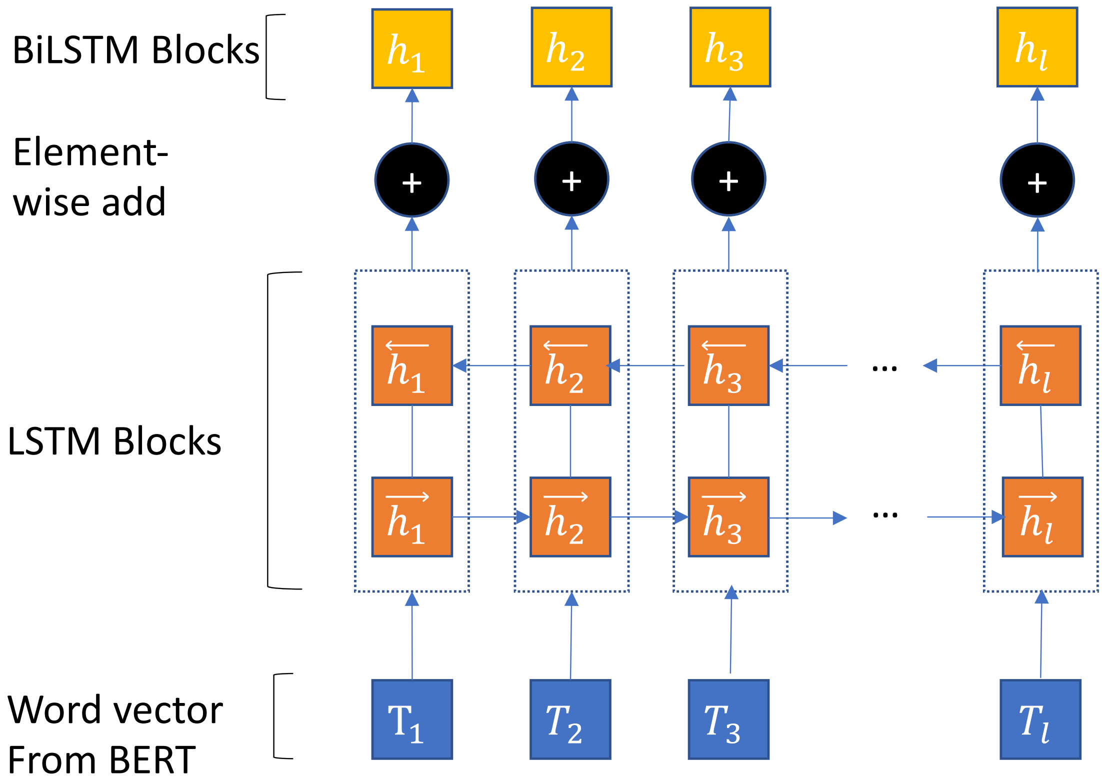


Fig. 2.8 Structure of Bi-LSTM

Human attention is a mechanism for quickly screening high-value information from massive information. The human attention mechanism inspires the deep learning attention mechanism. This method is widely used in various types of deep learning tasks such as NLP [**?**], image classification [**? ?**], and speech recognition [**?**], and has achieved remarkable results.

The essence of the attention mechanism is a group of key-value pairs $< K, V >$ contained in Source S, given an element of Query Q, which is followed by calculating the Q similarity to each K and remembering the weight coefficients of the corresponding values V. It is showed in Fig.**??**. Then a weighted sum is performed to obtain the final Attention value. It can be expressed as **??**, where Att and Sim represent attention and similarity.

$$Att(Q, S) = \sum_{i=1}^{|S|} Sim(Q, K_i) * V_i \tag{2.5}$$

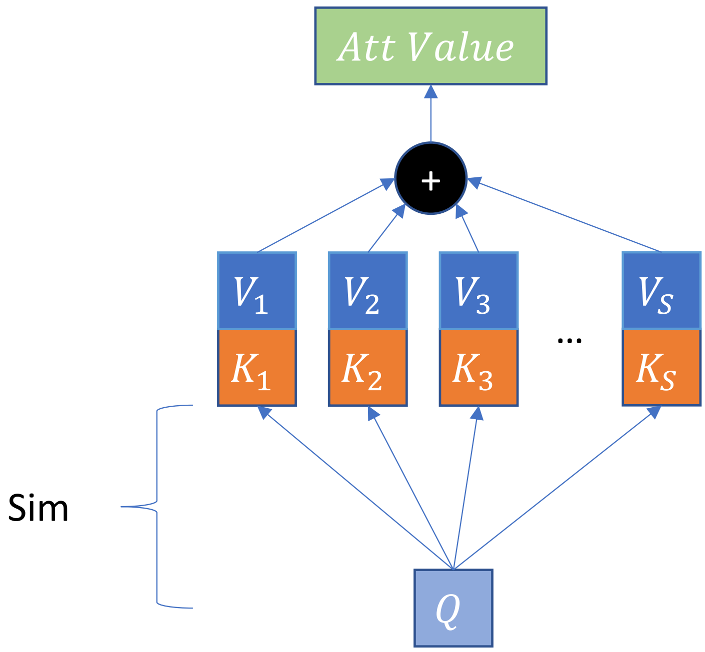


Fig. 2.9 The essential idea of Attention mechanism

Following the attention method, the encoder use Bi-LSTM and get hidden state vector $h_t = \overrightarrow{h_t} \bigoplus \overleftarrow{h_t}$. A word attention mechanism is introduced here. The core principle of the attention mechanism in this section is to compute the words that have the greatest influence on sentence meaning, i.e., the key semantic words, which are later aggregated into a vector of sentence meaning representations. The formula is **??**, where u_t is the hidden representation of h_t, r is the output text vector, and the u_ω is the similarity of the text vector of word aspect. By measuring the similarity between u_t and u_ω as the importance of the word, and using the softmax function to calculate normalization and to obtain the importance weight α_t. Finally, the entire text is transformed to word representation based on semantic contribution weights.

After that, the entire text is represented as a weighted sum of word annotations based on weights. The context vector u_ω can be regarded as a high-level representation of the fixed query "what is the word conveying information" and is randomly initialized as a learnable parameter in training.

$$\begin{aligned} u_t &= \tanh(W_\omega h_t + b_\omega) \\ \alpha_t &= Softmax(u_t^T u_\omega) = \frac{\exp(u_t^T u_\omega)}{\sum_t \exp(u_t^T u_\omega)} \\ r &= \sum_t \alpha_t h_t \end{aligned} \tag{2.6}$$

2.2.3 The GCN-based Knowledge Point Classifier Generator

In high school mathematics, knowledge points have more complex interrelationships such as correlation, subordination, inclusion, predecessor, successor, etc. These complex interrelationships are often difficult to model in Euclidean space, or this creates data sparsity problems. The establishment of relationships between knowledge points through graph data structures is more intuitive and has better interpretability. Recalling this model's task, it gives descriptions and answers to exercises, some of which have already marked knowledge points, and uses this information to mark knowledge points for exercises that are labeled knowledge points. Considering the dependence between knowledge points, some deeply hidden knowledge points that cannot be represented in shallow features can also be correctly labeled by the graph neural network output classifier. In this model, a GCN is used to learn and form the knowledge point connection graph, and each knowledge point corresponds to a node on the graph. After multiple graph convolution calculations, a series of classifiers are generated. These classifiers respectively act on the text vector generated by the text mining module. Each classifier outputs a value representing the probability that the knowledge point is associated with the exercise. Its overall structure is shown in the Fig.**??**.

Graph Convolutinal Network

In the real world, many important data sets generate connections as networks that form graph-like structures. Several papers reviewed this problem and attempted to generalize neural networks and apply them to arbitrary graph-structured data [**? ?**]. The GCN [**?**] is used

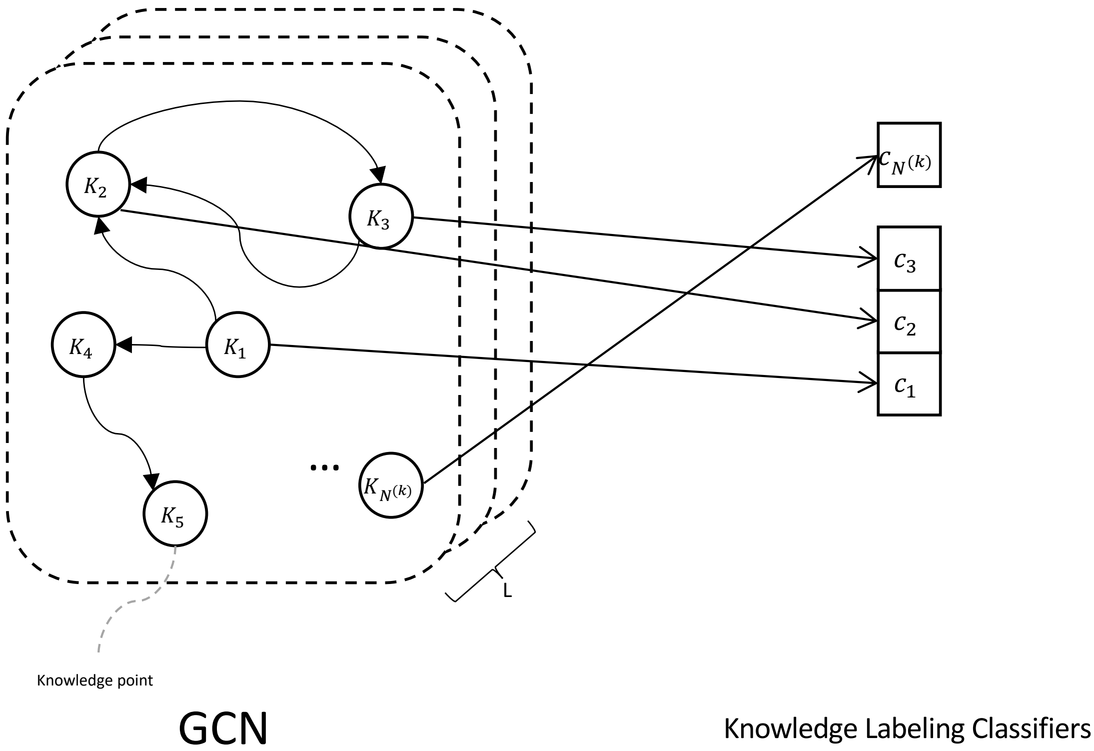


Fig. 2.10 The overview of structure

to process non-Euclidean spatial data that are difficult to learn by traditional convolutional neural networks. It is defined on a graph structure $G = (V, E)$, where V is the denote of vertex and E is the denote of edge inside the graph. The input of GCN $X = H^{(0)}$ is the first layer of GCN. The i-th layer is denoted as $H^{(i)}$, and $H^{(i)} \in \mathbb{R}^{N^{(v)} \times d_i}$, where $N^{(v)}$ is the denote of number of vertexes and d_i is the number of dimension of each vertex of layer i. The output layer $Z = H^{(L)}$ where L is the number of layers in GCN. Also, the GCN has another adjacency matrix A of dimension $N^{(v)} \times N^{(v)}$ used to describe the graph structure.

The core of graph convolutional learning is propagation, i.e., each node propagates information to neighboring nodes. The propagation of each layer is aggregated to form the next layer. The $i - 1$ layer $H^{(i)} \in \mathbb{R}^{N^{(v)} \times d_v}$ to layer $H^{(i+1)}$ transformation of GCN can be written in the formula **??**, where $f(\cdot)$ is a specific propagation method, e.g. all nodes spread their own values uniformly to neighboring nodes.

$$H^{(i+1)} = f(H^{(i)}, A) \tag{2.7}$$

A correlation matrix can represent the relationship between knowledge points, i.e., when a knowledge point i is related to a knowledge point j, the correlation matrix A models the knowledge point as a vertex in the GCN in order to learn the representation between knowledge points. The relation is learned by co-occurrence probability, i.e., supervised learning is performed on the library of exercises that have been tagged with knowledge points, which is based on the assumption that when multiple knowledge points have a high probability of occurring in an exercise, they should be intrinsically linked.

The node propagation of GCN in this section is **??**, where $\widehat{A}$ is the normalization form of $A \in \mathbb{R}^{N^{(v)} \times N^{(v)}}$. The $h(\cdot)$ is the nonlinear activation function LeakyReLU [**?**]. The parameter matrix $W^{(i)} \in \mathbb{R}^{d_i \times d_{i+1}}$ to be learned can be calculated by the statistical relations of the exercise knowledge points. The training process of Proposed is shown in Fig.**??**.

$$H^{(i+1)} = f(\tilde{A} H^{(i)} W^{(i)}) \tag{2.8}$$

Design of Correlation Matrix

In GCN, the correlation matrix A characterizes the relationship between graph nodes, and GCN information propagation calculation is also based on A. Therefore, designing the correlation matrix A is a crucial step in the GCN model. In this model, the commonly used data association rule mining algorithm Apriori algorithm calculates knowledge point association by knowledge point reference co-occurrence statistics.

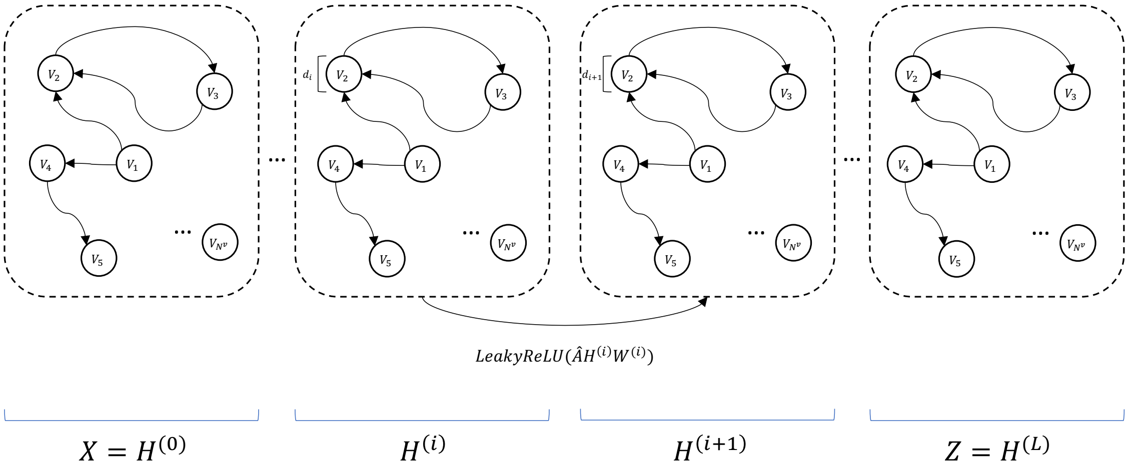


Fig. 2.11 The GCN training process

In this model, only the knowledge pairwise relationship needs to be found. Therefore, the knowledge point relationship matrix can be expressed as $R \in \mathbb{R}^{T \times T}$. The first task is to find the number of occurrences of frequently associated label in the label set. Support, Confidence, and Lift can be used to evaluate frequent label sets. Support is the proportion of the number of occurrences of label pair in the label set in the total label set. Confidence degree reflects the probability of a label $\mathscr{L}_i$ appearing, another label $\mathscr{L}_j$ appears, or the conditional probability of the data. Lift represents the probability that the label $\mathscr{L}_i$ is contained at the same time, and the ratio of the probability of occurrence of X population:

$$\text{Support}(\mathscr{L}_i, \mathscr{L}_j) = P(\mathscr{L}_i, \mathscr{L}_j) = \frac{\text{number}(\mathscr{L}_i, \mathscr{L}_j)}{\text{number(All Samples)}} \tag{2.9}$$

$$\text{Confidence}(\mathscr{L}_i \to \mathscr{L}_j) = P(\mathscr{L}_i \mid \mathscr{L}_j) = P(\mathscr{L}_i, \mathscr{L}_j)/P(\mathscr{L}_j) \tag{2.10}$$

$$\text{Lift}(\mathscr{L}_i \to \mathscr{L}_j) = P(\mathscr{L}_i \mid \mathscr{L}_j)/P(\mathscr{L}_i) = \text{Confidence}(\mathscr{L}_i \to \mathscr{L}_j)/P(\mathscr{L}_i) \tag{2.11}$$

Similar to calculating Support, the frequency matrix $E \in \mathbb{R}^{N^{(k)} \times N^{(k)}}$ of the sample knowledge point label pairs in the exercise training set can be calculated here. The M_{ij} represents the amount of co-occurrence between the knowledge point i and the knowledge point j in an exercise reference. Similarly, the knowledge point pair can be calculated by calculating Confidence to calculate the conditional probability matrix P, where $P_{ij} = P(\mathscr{L}_i, \mathscr{L}_j)/P(\mathscr{L}_j)$, where $P_{ij} = P(\mathscr{L}_i, \mathscr{L}_j)/P(\mathscr{L}_j)$ means the situation when the knowledge point j appears The conditional probability of the occurrence of the next knowledge point i.

It is a simple solution to directly set P as the incidence matrix A, but in actual situations, some comprehensive questions in the exercise set contain practically unrelated knowledge points, but these situations are relatively rare. In order to exclude the interference of accidental circumstances, a minimum knowledge confidence threshold can be set. When P_{ij} is greater than the given threshold $\tau^{(k)}$, naming activating value, then A_{ij} is set to P_{ij}, Otherwise A_{ij} is set to 0. The formula is like **??**.

$$A_{ij} = \{ \begin{array}{ll} 0, & \text{if } P_{ij} < \tau^{(k)} \\ P_{ij}, & \text{if } P_{ij} \geq \tau^{(k)} \end{array} \tag{2.12}$$

This model adopts the GCN structure because the parameter sharing of GCN for each node allows the learned classifier to retain the associated information in the knowledge association graph, thereby implicitly expressing its spatial semantic structure. Therefore, the output classifier can retain and identify the implicit knowledge label dependent information. For the dependence of knowledge points, refer to the data association method mining algorithm such as Apriori algorithm [**?**], and calculate the co-relation matrix by calculating the co-occurrence of knowledge points in the exercises.

Classifier generator

In this thesis, a learnable stacked GCN-based model is proposed. The knowledge point labels are represented by knowledge label word embedding. Knowledge point set $K = \{k_1, k_2, \ldots, k_{N^{(k)}}\}$, where $N^{(k)}$ denotes the amount of knowledge points. For each single knowledge point k_i, it is constrained that $k_i \in \mathbb{R}^{d^{(k)}}$, in which $d^{(k)}$ is the dimensionality of the embedding vector of knowledge point object. The knowledge label word embedding set is the input of GCN, i.e., $X = K$. The GCN is used to transform these knowledge point objects one by one into an internally connected knowledge point object classifier $C = [c_1, c_2, \ldots, c_{N^{(k)}}$, where $c_i \in \mathbb{R}^{d^{(r)}}$, $d^{(r)}$ is the dimensionality of the text representation vector r output by the text mining module. These classifiers and r can be used to calculate the dot product of each label. The structure overview is proposed in Fig.**??**.

2.2.4 Multi-label Recognition

From the stacked GCN network, we can learn interdependent object classifiers $C = \{c_1, c_2, \ldots, c_{N^{(k)}}\}$, where $N^{(k)}$ represents the number of knowledge points. Finally, the label prediction vector $\hat{y}$ can be obtained by the dot product of the learned classifier and the title text representation r:

$$\hat{y} = C \times r \tag{2.13}$$

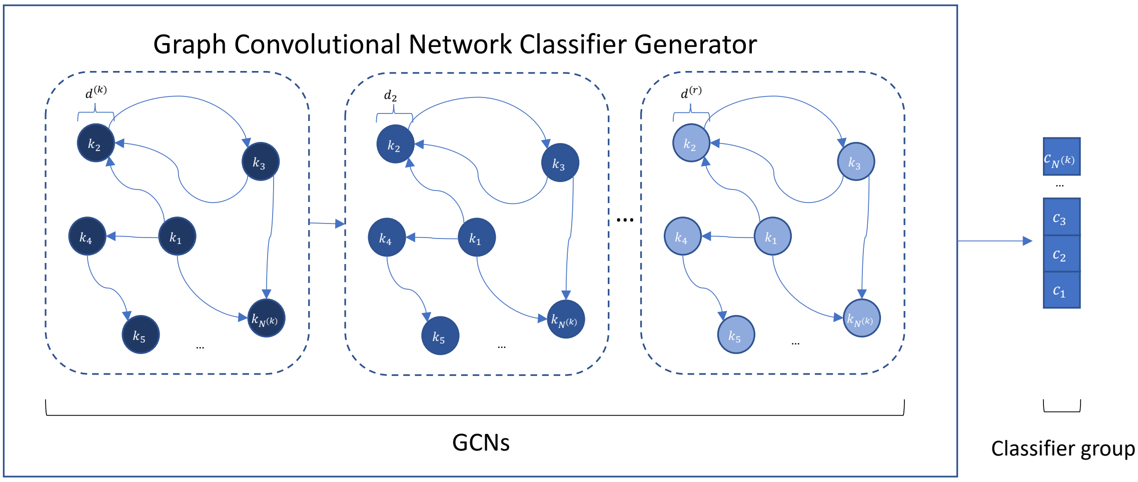


Fig. 2.12 The structure of GCN-based classifier generator

Through manual labeling, the real knowledge point labels of the exercises can be obtained: $y \in \mathbb{R}^C$, $y_i \in \{0, 1\}$, $y_i = 0$ means that the exercise does not have knowledge points The label of i, on the contrary, $y_i = 1$ means that the exercise has a label of knowledge point i. The loss function $\mathbf{L}$ can be written as:

$$\mathbf{L} = \sum_{i=1}^{T} y_i \log(\operatorname{sigmoid}(\hat{y}_i)) + (1 - y_i) \log(1 - \operatorname{sigmoid}(\hat{y}_i)) \tag{2.14}$$

2.3 Experiments

This chapter proposes a multi-label labeling model for exercise knowledge points. This section first introduces the data set, then introduces some baseline performance models, and then combines the multi-label labeling to propose a comparative performance indicator evaluation plan. Finally, Give comparison results and analysis.

2.3.1 Dataset

The experimental data comes from the labeled college entrance examination math test questions and simulation questions (including answer analysis) on the online website koolearn.com. The text corpus, such as question stems and answer analysis, are crawled through crawlers. Part of the knowledge point association model can be expressed as the following structure. The data set has been filtered and manually selected. There are 3374 test questions, including

148 knowledge points, with an average of 1.7 knowledge points. The distribution of several knowledge points is shown in the Fig.**??**.

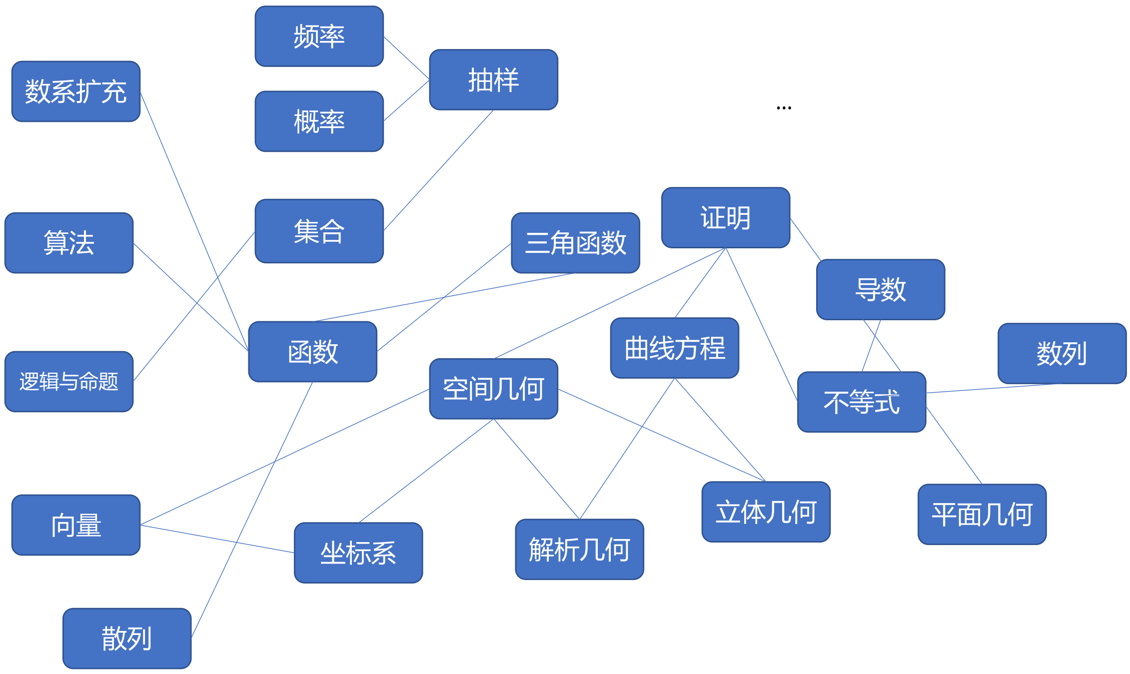
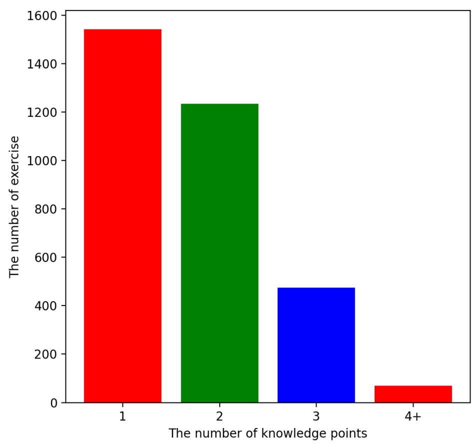
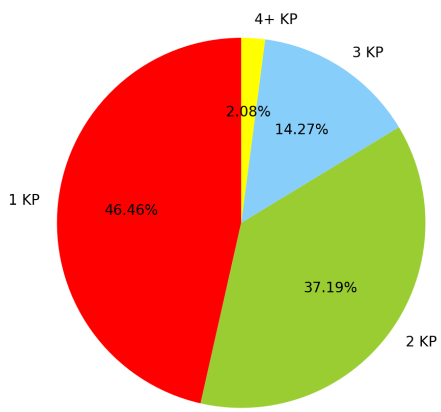


Fig. 2.13 Knowledge Points of dataset

The distribution of the number of exercise knowledge points in this data set is shown in the Fig.**??**.



(a) Number of Exercises



(b) Portion of Exercises

Fig. 2.14 Distribution of the number of knowledge points of exercise

2.3.2 Baseline

At present, there are already some algorithm models for labeling text. To verify the effectiveness of the algorithm proposed in this thesis, the following baseline models will be set as comparisons.

- Naive Bayes algorithm (NB): predict the labeling probability of a knowledge point based on the prior probability of text combination, and then convert the binary classification problem into a multi-label problem

- Multi-label KNN (ML-KNN): Proposed by Zhang et al. [**?**], it searches the k instances closest to the instance through the KNN algorithm, counts the number of each category in the k samples, uses the Naive Bayes algorithm to generate prediction outcomes for each label.

- CNN+word2vec: Use word2vec to convert the test question text into an embedded vector, extract the test text information through CNN, and then output the multi-label prediction

- CNN+BERT: Use BERT to convert the test question text into an embedded vector, extract the test text information through CNN, and then output the multi-label prediction.

2.3.3 Metrics

Compared with traditional learning problems, it is tough to label multi-label data, and multi-label means huge classification cost [**?**]. In the tasks in this section, the test questions need to be labeled with multiple knowledge points. It is a multi-label classification problem. The sample dimension, data volume, and label dimension will all affect the labeling effect. This section uses label-based metrics to evaluate the performance of the proposed test knowledge point labeling model.

Regarding multi-label classification, considering the relationship between the exercises' labels, this article uses label-based indicators for model evaluation. The indicator calculates the confusion matrix indicator for each label to calculate the sample values of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Then calculate the macro-F1 and micro-F1 metrics for multi-label tagging model performance evaluation.

According to the metrics proposed by Zhou et al. [?], for the j-th label of the i-th example in the n examples, shown in the formula **??**.

$$\begin{aligned}
\mathrm{TP}_j &= |\{x_i | \mathscr{L}_j \in Y_i \wedge \mathscr{L}_j \in \tilde{Y}_i, 1 \leq i \leq n\}| \\
\mathrm{FP}_j &= |\{x_i | \mathscr{L}_j \notin Y_i \wedge \mathscr{L}_j \in \tilde{Y}_i, 1 \leq i \leq n\}| \\
\mathrm{TN}_j &= |\{x_i | \mathscr{L}_j \notin Y_i \wedge \mathscr{L}_j \notin \tilde{Y}_i, 1 \leq i \leq n\}| \\
\mathrm{FN}_j &= |\{x_i | \mathscr{L}_j \in Y_i \wedge \mathscr{L}_j \notin \tilde{Y}_i, 1 \leq i \leq n\}|
\end{aligned} \tag{2.15}$$

where x_i and $\mathscr{L}_j$ represent the i-th exercise and j-th label, Y_i and $\tilde{Y}_i$ represent the real labels and predicted labels of the i-th exercise.

In the confusion matrix, Precision P is the proportion of data with correct predictions to the data that is predicted to be positive, and Recall R is the proportions of data with predictions that are positive to the actual data. F1 value is the harmonic mean value of precision rate and recall rate, and it is a comprehensive evaluation index of precision rate and recall rate. The calculation formula is as **??**:

$$\begin{aligned}
\mathrm{P} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \\
\mathrm{R} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\
\mathrm{F1}_{score} &= \frac{2}{\frac{1}{\mathrm{P}}+\frac{1}{\mathrm{R}}}
\end{aligned} \tag{2.16}$$

Similarly, for the multi-label tagging problem, Macro F1 and Micro F1 can be used as the metrics. Micro-F1 first calculates the total number of TP, FN, and FP, and then calculates F1, while Macro-F1 calculates the F1 of each category separately and then averages (the weight of each category F1 is the same). The formula is **??** and **??**.

$$\mathrm{F1}_{macro} = \frac{1}{c} \sum_{j=1}^{c} \mathrm{F1}_{score}(\mathrm{TP}_j, \mathrm{FP}_j, \mathrm{TN}_j, \mathrm{FN}_j) \tag{2.17}$$

$$\mathrm{F1}_{micro} = \mathrm{F1}_{score}(\sum_{j=1}^{c} \mathrm{TP}_j, \sum_{j=1}^{c} \mathrm{FP}_j, \sum_{j=1}^{c} \mathrm{TN}_j, \sum_{j=1}^{c} \mathrm{FN}_j) \tag{2.18}$$

Where c is the total number of labels.

Similarly, we can also count some indicators based on samples. For example, the multi-label accuracy rate Acc_{ML}, Hamming loss HmLoss, multi-label precision rate $\mathrm{Precision}_{\mathrm{ML}}$, multi-label recall rate $\mathrm{Recall}_{\mathrm{ML}}$ and $\mathrm{F1}_{ML}$ can be calculated. Accuracy is the proportion of samples with completely correct predicted labels in the overall sample. Hamming loss is a measure of the difference between the predicted label and the true label. Precision and Recall represent the proportion of true positive samples in true samples and the proportion of true positive samples in positive samples proportion. Their calculation formula is **??**-**??**.

$$\mathrm{Acc}_{ML} = \frac{1}{n}|\{i|Y_i = \tilde{Y}_i\}| \tag{2.19}$$

$$\mathrm{HmLoss} = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathrm{XOR}(Y_i, \tilde{Y}_i)}{c} \tag{2.20}$$

$$\mathrm{Precision}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap \tilde{Y}_i|}{|\tilde{Y}_i|} \tag{2.21}$$

$$\mathrm{Recall}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap \tilde{Y}_i|}{|Y_i|} \tag{2.22}$$

$$\mathrm{F1}_{ML} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{2.23}$$

2.3.4 Setting and Environment

The distribution of the number of exercise knowledge points in this data set is shown in the figure. The number of questions involved in different knowledge points is different. We define the set of questions involved in knowledge points j as $\mathbf{E}_j$. Remember the threshold of the number of occurrences of the knowledge point label $\tau^{(KP)}$, then the knowledge points of the exercises can be divided according to the frequency of occurrence, i.e., $\{j|\mathbf{E}_j| > \tau^{(KP)}\}$ and $\{j||\mathbf{E}_j| \leq \tau^{(KP)}\}$. According to the different thresholds, the model's ability to classify knowledge points that appear frequently and that appear sparse can be tested separately.

In the experiment, five sets of thresholds of 200, 150, 100, 50, 10 are set $\tau^{(KP)}$, and the exercises with the label j are recorded as $\mathbf{E}_j$, the label set that meets the requirements can be counted as $\mathbf{L}_\tau^{(KP)} = \{j \mid |\mathbf{E}_j| \geq \tau^{(KP)}\}$, the exercises containing label in $\mathbf{L}_\tau^{(KP)}$ is denoted as $\mathbf{E}_\tau^{(KP)}$, within which the average number of knowledge points of the exercise in the set is $\overline{L}$

Table 2.1 Setting of Experiment

$\tau^{(KP)}$	$\lvert\mathbf{L}_{\tau^{(KP)}}\rvert$	$\lvert\mathbf{E}_{\tau^{(KP)}}\rvert$	$\overline{L}$
200	2	463	1.21
100	22	1376	1.55
50	29	2237	1.42
10	57	3158	1.35

The running environment are shown in Table table:ch2-exp-env.

Table 2.2 Experiment Running Environment

Software/Hardware	Configuration
CPU	i7 9700K
GPU	Tesla V100
Operating System	Ubuntu 20.04
Python	3.8.6
PyTorch	1.6.0
GPU Driver	Cuda10.1/cudnn7

There are many adjustable parameters of the model in BERT and GCN. Hyperparameters are shown in Table **??**.

Table 2.3 Hyperparameter Settings of Recommendation Model

Hyperparameter	Value
Transformer Layers	12
Dimension of hidden layer	768
Optimizer	Adam
Learning Rate	0.001
LSTM dimension	150
batch size	32
Dropout	0.3

2.3.5 Result and Analysis

The comparison experiment results are divided into the performance comparison of the horizontal Baseline model (obtained in Table **??**-Table **??**) and the performance comparison of different hyperparameters of the model (Table 2.4).

It can be seen from the table that the performance of the model proposed in this thesis is stronger than the baseline model on the exercise training machine with a higher frequency. As the frequency of knowledge points decreases, the difficulty of classification increases, and performance degradation occurs in all models. The reason is that the model cannot effectively capture information for fewer tags, resulting in increased errors. In general, the model proposed in this thesis has achieved the best or better performance in terms of F1-Score parameters and stricter subset accuracy indicators. When the frequency of the problem labels is very low, almost all models cannot achieve good results. This is because due to the

Table 2.4 Result comparison ($\tau^{(KP)} = 200$)

Metrics	$F1_{macro}$	$F1_{micro}$	Acc_{ML}	HmLoss	$F1_{ML}$
NB	75.3	74.2	69.6	18.2	73.6
ML-KNN	77.1	76.2	73.2	17.4	76.3
CNN+word2vec	79.5	78.4	76.6	14.2	79.6
CNN+BERT	80.1	**79.9**	76.9	13.7	79.5
Proposed	**80.9**	79.1	**77.3**	**13.1**	**80.7**

Table 2.5 Result comparison ($\tau^{(KP)} = 100$)

Metrics	$F1_{macro}$	$F1_{micro}$	Acc_{ML}	HmLoss	$F1_{ML}$
NB	71.2	72.1	67.2	16.2	71.8
ML-KNN	73.2	72.3	69.1	15.9	74.7
CNN+word2vec	74.3	74.4	72.3	13.2	75.2
CNN+BERT	74.4	74.6	72.3	13.1	75.1
Proposed	**75.5**	**75.7**	**73.1**	**12.7**	**74.9**

Table 2.6 Result comparison ($\tau^{(KP)} = 50$)

Metrics	$F1_{macro}$	$F1_{micro}$	Acc_{ML}	HmLoss	$F1_{ML}$
NB	52.3	53.0	42.1	9.2	51.9
ML-KNN	44.2	43.9	23.5	10.1	42.1
CNN+word2vec	56.1	**57.3**	46.2	8.2	56.5
CNN+BERT	56.2	56.8	**47.0**	**8.1**	56.1
Proposed	**57.1**	57.2	45.2	8.6	**57.5**

Table 2.7 Result comparison ($\tau^{(KP)} = 10$)

Metrics	$F1_{macro}$	$F1_{micro}$	Acc_{ML}	HmLoss	$F1_{ML}$
NB	36.5	37.1	26.1	4.2	36.5
ML-KNN	30.1	32.1	29.1	3.6	32.5
CNN+word2vec	36.7	38.2	37.5	3.5	37.2
CNN+BERT	36.9	**38.6**	**38.6**	**3.5**	37.5
Proposed	**37.1**	38.3	35.4	3.8	**38.6**

uneven frequency distribution of the problem labels, a small number of unpopular labels cannot be well labeled, resulting in over-fitting in model training. This phenomenon affects labeling performance. In order to solve this problem, the model prediction performance can be optimized by using a larger set of exercises with a more average frequency of knowledge points than the training set.

2.4 Summary

In the exercise recommendation system, there are a large number of exercises with missing knowledge point labels. To meet the requirements of the adaptive learning system, it is necessary to label the exercises, but the manual labeling cost is high, and the efficiency is low. Therefore, automatic marking of knowledge points of test questions has become an urgent problem to be solved. This chapter proposes a multi-knowledge point labeling model for exercises based on GCN and Bi-LSTM text mining model based on the attention mechanism. After verifying the experimental data set, a better knowledge point annotation effect than existing models has been achieved.

The contributions of this chapter are as follows:

1. The relationship between knowledge points is represented by a graph neural network, which can dig out the hidden knowledge point labels in the original text and provide the existing model with the knowledge point label association reasoning function.
2. By designing the correlation function between the knowledge points, the dependence between the knowledge points is modeled, and good performance has been achieved.
3. It is verified that the low-knowledge point label frequency will produce an over-fitting phenomenon for the multi-label classification model, resulting in performance degradation. To solve this problem, it can be solved by averaging the label frequency of the training data set.

The labeling of exercise knowledge points is the first step of the recommendation system. The labeled exercises can be used as the input of the knowledge tracking model to track the knowledge state of students and can also be used as the input feature of the recommendation system to complete the recommendation of exercises based on knowledge points.