Writing your thesis in LaTeX2e

Wangzhihui Mei

Supervisor: Zhifeng Wang

This thesis is submitted for the degree of

Master of Engineering

Central China Normal University Wollongong Joint Institute

Central China Normal University

February 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
February 2021

Abstract

is paper implements a high school mathematics learning resource recommendation system based on knowledge tracking and factorization machine, which contains three core points. The first one is the design of data source, and proposes an automatic extraction method for test knowledge points. The whole process of high school mathematics knowledge mapping from corpus collection, construction, storage, and update is introduced as well as the automation process. The second point is the implementation of a GCN-based knowledge tracking algorithm, which takes into account the a priori structure of knowledge points as well as students' recent question records and knowledge mastery changes, and achieves a more excellent performance compared to other knowledge tracking models. The third point is based on a deep factorization machine, which solves the problem of sparsity of training data, and it takes the output of the knowledge tracking model as input and considers various other feature inputs to achieve learning resource recommendation.

Keywords: Learning Resource Recommendation System, Knowledge Graph，Knowledge Tracing，Factorization Machine，T，h

摘 要

中文摘要...

关键词：keyword1，keyword2，keyword3，keyword4

Acknowledgments

Publications

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

目录

插图

表格

Nomenclature

Subscripts

crit Critical state

Acronyms / Abbreviations

ALU Arithmetic Logic Unit

BEM Boundary Element Method

CD Contact Dynamics

CFD Computational Fluid Dynamics

CK Carman - Kozeny

DEM Discrete Element Method

DKT Draft Kiss Tumble

DNS Direct Numerical Simulation

EFG Element-Free Galerkin

FEM Finite Element Method

FLOP Floating Point Operations

FPU Floating Point Unit

FVM Finite Volume Method

GPU Graphics Processing Unit

LBM Lattice Boltzmann Method

LES Large Eddy Simulation

MPM Material Point Method

MRT Multi-Relaxation Time

PCI Peripheral Component Interconnect

PFEM Particle Finite Element Method

PIC Particle-in-cell

PPC Particles per cell

RVE Representative Elemental Volume

SH Savage Hutter

SM Streaming Multiprocessors

USF Update Stress First

USL Update Stress Last

Chapter 1

Introduction

1.1 Research Background and Significance

Artificial intelligence industry has developed rapidly in recent years, it is being commercialized in all aspects, triggering profound changes in various industries, and the future development of artificial intelligence will be the combination of key technologies and industries.[3] At present, AI technology has been implemented in many fields such as finance, medical and security, and the application scenarios are becoming more and more abundant. The commercialization of AI has played a positive role in accelerating the digitization of enterprises, improving the structure of the industrial chain, and increasing the efficiency of information utilization. The traditional education industry also tries to use AI technology to help the development of the industry. Every development of AI is accompanied by breakthroughs in research methods, and deep learning is one of the important representatives of the breakthroughs in machine learning technology in recent years. With the continuous extension of human AI research and application fields, AI will usher in more kinds of technology combination applications in the future. Artificial intelligence has also begun to be applied to the education industry, and the concept of intelligent education has emerged. Among the types of applications of AI technology in education, AI adaptive learning is the most widely used in all aspects of learning. In addition, due to China's large population base, the shortage of educational resources, the importance attached to education and other favorable factors intelligent adaptive learning system is expected to come later.

In recent years, domestic adaptive learning has begun to enter the minds of many people involved in education training and education investment. There are more and more education

technology companies in the market that focus on adaptive learning tools. At the same time, many education companies have started to use adaptive learning as the main core function or main selling point of their products. The biggest advantage of adaptive education is that it can locate the knowledge gaps of each student. The adaptive learning platform will guide the student to the next most suitable learning content and activities for him. When students encounter a course that is too difficult or too low in the learning process, they can automatically adjust the difficulty of the course. Teachers can also analyze the knowledge gaps of each student based on the learning status evaluation report provided by the system, adjust the learning progress in real time, and provide personalized teaching for each student. So theoretically, adaptive learning is one of the potentially feasible solutions to the problem of "teaching to students according to their abilities" in online education. To make a practical adaptive learning system, I plan to use knowledge tracing to track students' learning status and use the factorization machine algorithm to calculate the relevance of topics to students to build such a test recommendation system.The current personalized learning resource recommendation system is one way of implementing adaptive learning, which is the subject of this paper.

In this paper, the study focuses on the recommendation of learning resources for the subject of high school mathematics. In this system, there are two aspects in general: on the one hand, scientific and targeted acquisition and tracing of students' knowledge state, and on the other hand, recommendation of personalized learning resources based on students' knowledge mastery state. We use the knowledge tracing algorithm of graph neural network to acquire and track students' knowledge states, and the factorization agent to try to combine the output of graph neural network with prior knowledge for resource recommendation.

1.2 Research Status

The dominant content of the research is knowledge tracing and recommendation system. Some advanced graph neural network algorithm is applied to finish the task. There have been some research advances and related applications in the area of knowledge tracing and factorization. We surveyed some existing knowledge tracing algorithms and applications, and some applications of factorization machine.

1.2.1 Property of high school Math

Disciplines and knowledge are closely related to each other, so that disciplinary knowledge denotes the specific knowledge contained in a particular field of study. Disciplines are referred to in this study only for specific subjects in the field of education, such as mathematics, language, chemistry and so on. The first step is to learn how to make the best use of the knowledge that is available. The knowledge is obtained from practice, so after learning it, it can also be applied to social practice. Scientific knowledge is declarative because it can be expressed in a series of symbols, words and diagrams; it is also procedural because it can be arranged and learned according to a specific logical order in the process of concrete learning.

Mathematics is a science specializing in the study of the relationship between quantities and spatial forms, its symbolic system is more complete, the formula structure is clear and unique, text and images and other expressions of language is also more vivid and intuitive.

The knowledge that learners need to learn mostly comes from the summaries of the experiences of their predecessors in practical activities. The learning process is a process of cognitive learning of the summarized knowledge and continuous digestion, adjustment and reorganization of the knowledge structure, so as to build a more perfect and suitable knowledge structure, as well as a process of integration with innovative thinking. Thus a good cognitive structure can promote the formation of knowledge structure, and a good knowledge structure can enrich the organization form of cognitive structure. Since the disciplinary knowledge structure consists of two parts: knowledge composition and knowledge dependency, we will analyze the disciplinary knowledge structure from these two aspects, knowledge structure and composition.

The knowledge that learners need to learn mostly comes from the summaries of the experiences of their predecessors in practical activities. The learning process is a process of cognitive learning of the summarized knowledge and continuous digestion, adjustment and reorganization of the knowledge structure, so as to build a more perfect and suitable knowledge structure, as well as a process of integration with innovative thinking. So a good cognitive structure can promote the formation of knowledge structure, and a good knowledge structure can enrich the organization form of cognitive structure. Since the disciplinary knowledge structure consists of two parts: knowledge composition and knowledge dependency, we will analyze the disciplinary knowledge structure from these two aspects.

Knowledge composition refers to the organization of knowledge within a subject area, which mainly includes knowledge points, knowledge blocks and knowledge systems.

- knowledge point: A point of knowledge is the smallest constituent unit of the knowledge structure of a discipline and is used to represent specific concepts.
- knowledge block: A knowledge block is a collection of one or more sets of knowledge points, also known as knowledge modules, in which knowledge blocks and knowledge blocks can be combined to form new knowledge modules, and a subset of knowledge blocks is called a knowledge sub-module.
- knowledge body: a body of knowledge is a structured system that is a combination of all the pieces of knowledge in a particular subject area.

Mathematics is a science that specializes in the relationship between quantity and spatial form. Its symbol system is more complete, the formula structure is clear and unique, and the language of expression such as words and images is more vivid and intuitive. The knowledge structure of senior secondary mathematics is a more logical and systematic knowledge system organized on the basis of the knowledge structure of junior secondary mathematics. This is because learning for any discipline needs to be based on the existing cognitive structure in order to progressively effective learning and skills training, so in the process of learning high school mathematics, you need to have a solid foundation of junior high school mathematics discipline knowledge. In the past few years, there have been a number of cases in which the students have been able to learn from each other.

- Highly abstract: Mathematics has a high degree of abstraction, because the discipline's knowledge system is built using many abstract knowledge concepts, and with the help of these concepts and knowledge to learn and expand thinking, forming new abstract conceptual knowledge. The abstraction of mathematics is reflected in the object is not concerned with the introduction of specific content, only the number of relationships between the spatial form. Therefore, abstraction in mathematics is different from abstraction in other disciplines in terms of both object and degree. There are also some differences between mathematics and the natural sciences, because in mathematics the accuracy of calculations, proofs, and inferences can only be verified using rigorous logical methods and cannot be tested by repetitive experiments, whereas in the natural sciences the verification is the opposite.

- Strict logic: The discipline of mathematics is very logical because any conclusion reached in mathematics requires rigorous logical reasoning and rigorous proof in order to be considered reasonable. However, mathematics is not the only discipline that possesses rigorous logic; other natural science studies of reasoning and proof must also possess a certain degree of logic. In mathematics, not all conclusions reached after reasoning and proof can be applied in practice, because many mathematical models are developed and mathematical conclusions drawn under ideal circumstances.

- Broad applicability: Mathematics is an important means and tool for us to participate in practical social activities or scientific research, and the study of mathematics is indispensable in all walks of life and in all areas of society. Therefore, mathematics has a wide range of applications and has become an important basis for the development of modern science.

1.2.2 Knowledge relation

Knowledge relations represent the connections between knowledge points (or between knowledge blocks and knowledge chunks) in the discipline knowledge structure. It is through these connections that different knowledge points can be formed into knowledge blocks, and different knowledge blocks can be combined to form the whole disciplinary network knowledge structure system. There are many different kinds of knowledge relationships, so that different definitions of knowledge relationships lead to different knowledge structures. Therefore, in order to unify the definition of knowledge relations, we divided them into general relations and special relations based on the general and special characteristics of discipline knowledge structure. The special relationships represent the unique knowledge relationships of a particular discipline, while the universal relationships represent the general relationships of any discipline. Secondly, according to the demands of knowledge graphping research, we divide universal relations into six kinds of knowledge relations: synonymous, fraternal, antecedent, consequent, inclusive and antagonistic; and special relations into four kinds of knowledge relations: detailed, transformative, causal and correlative.

- tautology: Expresses the relationship between two points of knowledge that have the same meaning as what is being described, e.g. regular and equilateral triangles.

- fraternity: Expresses the relationship between two knowledge points that have the same parent class.

- predecessor: It means that you need to finish learning knowledge point A before learning knowledge point B, that is, $A \rightarrow B$ is a precursor relationship.

- successor: denotes the inverse of the antecedent relationship, i.e., $B \rightarrow A$ is the successor relationship.

- containment: Indicates that knowledge point B is included in the definition of knowledge point A, i.e., $A \rightarrow B$ is an inclusion relationship.

- antagonism: From a certain point of view, knowledge point A is incompatible with knowledge point B, i.e. $A \leftrightarrow B$ is an antagonistic relationship.

- refinement: A grammatical analysis of the definition of knowledge point A leads to knowledge point B, where $A \leftrightarrow B$ is a detailed relationship

- transformation: denotes that knowledge point A and knowledge point B can be transformed to each other under certain conditions, i.e., $A \leftrightarrow B$ is a transformation relationship.

- causation: denotes that knowledge point A can be deduced from knowledge point B as a known condition, i.e., $A \leftrightarrow B$ is a causal relationship.

- relation: Indicates that there is a relationship between the definitions of Knowledge Point A and Knowledge Point B, but the relationship is not explicitly specified, i.e., $A \leftrightarrow B$ are correlated.

In the process of constructing the discipline knowledge structure, firstly, we need to analyze the current discipline knowledge content, teaching objectives, teaching objects, teaching strategies and discipline characteristics in detail; secondly, we divide the whole discipline knowledge system into several knowledge modules, and then we divide each knowledge module into several knowledge points; finally, with reference to the above ten kinds of knowledge relationships and the knowledge relationships extracted from data sources, we can determine and establish the relationships between knowledge modules and knowledge modules, between knowledge modules and knowledge points, and between knowledge points and knowledge points, so as to form a complete discipline knowledge system structure.

1.2.3 Knowledge tracing algorithms

Knowledge Tracing is a technique that models students' knowledge acquisition based on their past answers to obtain a representation of their current knowledge state. The task is to automatically track the change of students' knowledge level over time based on their historical learning trajectory, in order to be able to accurately predict the students' performance in future learning and to provide appropriate learning tutoring. In this process, the knowledge space is used to describe the level of student knowledge acquisition. A knowledge space is a collection of concepts, and a student's mastery of a part of a collection of concepts constitutes the student's mastery of knowledge. Some educational researchers argue that students' mastery of a particular set of related knowledge points will affect their performance on the exercise, i.e., the set of knowledge that students have mastered is closely related to their external performance on the exercise.

The task of knowledge tracing is to model the student's knowledge mastery state based on the student's answer record, which is usually a time series, and in some business scenarios is time-independent, so that we can accurately predict their future answers and make reference for future intelligent questioning based on this to avoid giving students too difficult or too easy questions. Specifically, suppose a student's answer record is $x_0, x_1, ..., x_t$, and we are going to predict the next interaction x_{t+1}, usually one interaction $x_t = (q_t, a_t)$, q_t represents the right or wrong situation of that student's answer to the question a_t.

There are several kinds of knowledge tracing algorithms:

- Bayesian knowledge tracing(BKT): Bayesian knowledge tracing is an early and commonly used knowledge tracing model, BKT uses user interaction modeling with real-time feedback to model a learner's potential knowledge state as a set of binary variables, each representing whether or not a knowledge point is understood, and there are dynamic changes in mastery of knowledge points as students continue to practice, BKT maintains binary variables of knowledge point proficiency by using Hidden Markov Models (HMM), the original BKT model does not take into account students' knowledge forgetting, and related studies address students' guesses, personal vivid knowledge mastery and problem difficulty factors on BKT[10].

- Deep Knowledge Tracing(DKT): The DKT model applies neural networks to the knowledge tracing task for the first time[6], using an LSTM model to track the dynamics of student knowledge proficiency over time, and to learn the potential vector

representation of student knowledge proficiency directly from the data. The advantage of DKT is that it can record knowledge over a longer period of time based on students' recent answers. In addition, it can update the knowledge state based on each answer, only the last implicit state needs to be saved, no double counting is required, and it is suitable for online deployment. It does not require domain knowledge, works with any user answer dataset and automatically captures associations between similar questions. The disadvantage of DKT is that the model output fluctuates greatly when the answer sequence is disrupted, i.e., the same questions and the same responses yield different knowledge states when the answer sequence is inconsistent. Due to the above-mentioned problems and the fact that students do not necessarily have continuous consistency in their k nowledge during the answer process, it leads to bias in the prediction of students' knowledge states influenced by the sequence. There is also the black box problem, which sometimes leads to the strange situation that the first correct answer leads to a high prediction probability for all subsequent ones, while the first wrong answer leads to a low prediction probability for all subsequent ones.

- Dynamic Key-Value Memory Networks for Knowledge Tracing(DKVMN): Dynamic Key-Value Memory Networks for Knowledge Tracing (DKVMN) was proposed in 2017 by Jian Jian of the Chinese University of Hong Kong[11]. Based on the strengths and weaknesses of BKT and DKT and using the memory augmentation neural network approach, the Dynamic Key-Value Memory Networks (DKVMN) is proposed. It borrows ideas from memory-enhanced neural networks and combines the advantages of BKT and DKT. DKVMN stores all knowledge points with a static matrix key and a dynamic matrix value to store and update the student's knowledge state. In the DKVMN paper, they compare DKVMN with DKT and a sophisticated version of BKT, BKT+. They found that DKVMN achieves excellent performance and is the most advanced model in the KT domain. In addition to improved performance, it has several other advantages over LSTM, including prevention of overfitting, a smaller number of parameters, and automatic discovery of similar practice questions by underlying concepts. In addition, Chaudhry R[2] improves the performance of DKVMN by jointly training request cue prediction with knowledge tracing through multi-task learning.

1.2.4 Factorization Machine

The factorization machine model is a factorization model that can be used in large scale sparse data scenarios[7]. The solution of this model is linear in time complexity and he can solve it directly using raw data without relying on support vectors like SVM. In addition, FM is a general model that can be used on any real data and can do tasks such as classification and regression and even sorting. The idea is that the idea is to add a linear combination of two features to linear regression, and the way to solve the linear combination is to use a matrix decomposition. FFM is an improvement on FM by adding the concept of Field[5], that is, the class to which each feature belongs. Suppose there are f Fields. Then each feature has to have f hidden vectors. When two features are crossed, the dot product of each feature and the vector corresponding to the other Field is used. In this way, it is ensured that the same Field does the same thing for the same feature, and the features of different Fields do different things for the same feature. In addition, there is also a deep learning version of the factorization machine algorithm[4], and this model is improved based on wide and deep. First, the model includes FM and DNN parts, which is a parallel structure, and FM and DNN share the same input (embedding). The mapping vector from the field to the embedding layer is exactly the vector learned by the FM layer. It has the advantage that it does not require pre-training and can learn the intersection of low and high dimensional features.

1.3 Research Objectives and Content

The purpose of this study is to build a high school mathematics learning resource recommendation system based on knowledge tracing and factorization machine algorithm. We use knowledge tracing to model students' knowledge states, which outputs a graphical knowledge state vector, which we use as the next-level input, considering students' individualized differences and knowledge forgetting process, and apply the factorization machine algorithm to the resource recommendation system. For knowledge tracing, we build a graph neural network-based knowledge tracing model, which can well characterize the intrinsic connections of knowledge points in mathematics subjects considering that the knowledge points are a graph-like structure, and output a graph knowledge vector matrix, which can also effectively characterize the connections between problems and knowledge points. The output of the knowledge tracing model is then passed through a factorization

machine algorithm to obtain the recommendation degree of the learning resources and output a vector of recommendation weights for different learning resources.

1.4 Thesis Organization and Structure

Chapter 1 of this paper is an introduction. It introduces the research background of the study, current industry-related research progress and the focus of the study. Then it leads to the three core points of this paper: learning resource representation, knowledge tracing and resource recommendation.

Chapter 2 of this paper concentrates on learning resource representation, which addresses storing learning resources through knowledge graphs. This paper explores some concepts of knowledge graphs, related studies, and then gives the process of knowledge graph building. It is also demonstrated that knowledge graph building can effectively characterize the a priori intrinsic features of subject knowledge.

Chapter 3 of this paper gives a knowledge tracing model of pre-trained graph neural network, which better characterizes the graph-like properties of knowledge. It is able to transform the knowledge tracing task into a time-series node-level classification problem in GNNs. Since the knowledge graph structure is not explicitly provided in most cases, we present various implementations of the graph structure. Empirical tests on two open datasets show that the method improves the prediction of student performance without any additional information and shows more interpretable predictions. The inclusion of pre-training is also attempted during the experiments, which can greatly improve the training efficiency and performance.

Chapter 4 of this paper proposes the application of a deep factorization machine algorithm with knowledge tracing model data as input to build a learning resource recommendation system. The output is a weight vector of top n, characterizing the recommended resources of top n.

Chapter 5 of this paper presents the conclusion.

Chapter 2

Learning Resource Database based on Knowledge Graph

2.1 Research Motivation

As the starting point of a recommendation system, the data source is often the first step to consider. b A knowledge graph is a map that serves as a kind of knowledge domain mapping, which shows the knowledge development process and structure. Visualization techniques are used to describe knowledge resources and their carriers, to mine, analyze, construct, map and display knowledge and their interconnections. Also knowledge graphs, as an important part of artificial intelligence, are now playing an increasingly important role in recommendation systems and question and answer systems.

In this section, this paper uses the knowledge graph to store the knowledge points and questions of high school mathematics as the data source for the recommendation system. It serves as a supplementary input to the knowledge tracing model section and provides a priori k中nowledge of the subject knowledge points.

Firstly, the basic principles and common techniques of knowledge graph are introduced, and then a knowledge graph construction method for automatic extraction of knowledge points of test questions is proposed, and the knowledge graph construction and knowledge point extraction, corpus acquisition and preprocessing, knowledge point identification and knowledge graph updating techniques are introduced.

In this paper, we will use the knowledge graph in constructing the knowledge graph of mathematics domain and building the student user model. In the construction of the

knowledge graph of high school mathematics domain, the basic concepts and keywords involved in the textbooks are abstracted into knowledge point ontologies, the knowledge point keywords in high school mathematics exercises are identified by the named entity recognition method, and then the abstracted knowledge points are associated with the keywords to build the knowledge graph of high school mathematics. A large number of knowledge points in high school mathematics are associated with the degree of mastery of each knowledge point by individual students, which is built into a student user model, and the abstraction of the student user model into a knowledge graph will facilitate the personalized recommendation of exercises.

2.2 Research Status

2.3 Basic theory of knowledge graph

The Knowledge Graph is a new concept introduced by Google in 2012. Knowledge graph is essentially a knowledge base of Semantic Network, which describes conceptual entities and their relationships in the objective world in a structured form. From the beginning of oogle search, to nowadays chatbots, big data risk control, securities investment, intelligent medical care, adaptive education, recommendation system, all use knowledge graph.

2.3.1 Representation

Knowledge graphs focus on concepts, entities and their relationships, where entities are things in the objective world and concepts are generalizations and abstractions of things with the same properties. Ontology is the basis of knowledge representation of knowledge graph, which can be formally represented as $O = \{C, H, P, A, I\}$, where C is the set of concepts, such as transactional concepts and event-like concepts, H is the set of contextual relations of concepts, P is the set of attributes, which describes the features possessed by concepts, A is the set of rules, which describes the domain rules, and I is the set of instances, which describes the instance-attribute-value.

The common knowledge graph representations are Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), Web Ontology Language (OWL), etc.

1. resource description framework RDF is the most commonly used symbolic semantic representation model, which provides a unified standard for describing entities/resources. the basic model of RDF is a directed labeled graph, each edge on the way corresponds to a subject-predicate object triad, and a triad corresponds to a statement of an event. the RDF consists of nodes and edges, the nodes represent entities/resources, attributes, and the edges represent entities and the relationship between entities and attributes. Figure 2.1 shows the relationship between entities and attributes.

2. lightweight schema language RDFS is an extension of RDF by adding the definition of class properties and other Schema layers on top of the objective events provided by RDF. rdfs is mainly used to define term sets, class sets and property sets, mainly including classes, subclasses, properties, subclass properties, domains, scopes and other primitives, which can build the basic class hierarchy and These clauses can build the basic class hierarchy and attribute system.

3. OWL is the core of the Semantic Web technology stack, which provides fast and flexible data modeling capability and efficient automatic reasoning capability.

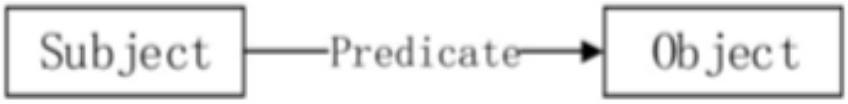

Fig. 2.1 Structure of Triad

2.3.2 Storage

There are mainly two kinds of storage for knowledge graphs, one is based on the academic idea of RDF, which is based on triples car storage: the other is based on the industrial idea of attribute graph, which is based on attribute graph storage.

Using RDF for storage, mainly RDF serialization methods, such as RDF/XML, N-Triples, Turtle, RDFa, JSON-LD, etc.

The most important feature of the graph database for storing knowledge graphs is that the graph database can store not only the nodes and relations of triples, but also the attributes of their nodes and relations together, and the graph can be traversed efficiently, which is convenient for publishing data and retrieval of knowledge graphs.

2.4 The construction of knowledge graph

In order to convert a large number of standard questions into feature questions in a timely manner, this paper proposes an automatic knowledge extraction method based on knowledge graph for test questions. The method uses natural language processing techniques such as named entity recognition, keyword extraction and calculation of semantic similarity of keywords to collect knowledge points and mathematics-related vocabulary in high school mathematics domain and build up a knowledge map of high school mathematics. The mathematics test questions are divided into words and entities by customized dictionaries and rules, and the candidate keywords are extracted and brought into the knowledge map of high school mathematics for querying.

2.4.1 The extraction of knowledge point

In this paper, the architecture for building the knowledge map of high school mathematics is mainly divided into four modules: corpus collection module, pre-processing module, knowledge map building module and knowledge extraction system module. The high school mathematics mapping consists of two entities: mathematical knowledge points and mathematical vocabulary related to mathematical knowledge points, which are mainly used to provide automatic knowledge extraction function for the construction of intelligent question bank later. The question bank not only contains a large number of mathematical knowledge points, but also contains a lot of mathematical vocabulary related to mathematical knowledge points. The mathematical vocabulary does not only stop at the mathematical concept ontology, mathematical textbook chapter names, mathematical knowledge points, etc., but also includes words or vocabulary that can deduce the knowledge points of the test questions. For example, if the operation symbol of "$\cap$" appears in the question, it can be deduced that the question is probably about "set", "set operation", "set intersection operation", "set element interdependence", "set interval", "set interval", "set interval", "set interdependence", "set interval", etc. "The pre-processing module is to standardize the imported data corpus to form structured data, and then to identify the named entities, which mainly involves the process of entity identification and relationship extraction. High school mathematics is relatively a semi-closed and semi-open subject, the mathematical knowledge is relatively closed, but the mathematical test questions are very different, but the final solution still comes back to the test mathematical knowledge, so it is semi-open. The relevant mathematical knowledge

points and concepts obtained through knowledge extraction need to be evaluated by experts to create a valuable and credible knowledge map. The knowledge extraction system plays a key role in the construction of the intelligent question bank by tagging the questions obtained by crawlers on the web after processing.

2.4.2 Corpus data preprocessing

After importing the textbook and web crawler corpus, although a large amount of valuable text, image information and structured data such as question stems and parses in the question bank have been obtained, there are a large number of mathematical formulas in these data, and these mathematical formulas are not standardized because of the platform, and there is no uniform storage format, so it is necessary to standardize these text data containing mathematical formulas.

- standardization: The standardization process is mainly for the pictures and mathematical formulas in the corpus, many crawlers get the test information through the pictures, the test information stored in the pictures is very unfavorable to the word separation and naming entity recognition, and the mathematical formulas also exist in a variety of storage methods, currently more common storage and display of mathematical formulas are Office comes with the formula editor plug-in, Mathtype formula editor, Mathml, Latex and so on. For picture information, we use OCR technology for text recognition, and for mathematical formulas, we use Latex format. Mathpix is a software for quickly recognizing mathematical formulas stored in pictures and converting them into Latex format with high recognition accuracy and efficiency, and Latex can extract some useful semantic information for later knowledge extraction when expressing mathematical formulas.

- User dictionaries and stopwords: User dictionaries, also known as user-defined dictionaries, are mainly used to enhance the disambiguation and error correction ability of the subscripts by manually adding subscripts rules in the process of named entity recognition, and the subscripts recognition will give priority to the words in the user dictionaries for subscripts after adding the user dictionaries. At present, there is no mature lexical corpus in the field of mathematics, so it is important to build a user lexicon related to mathematics. For example, if we do not add a user lexicon for "function analytic", we will get [function, analytic, equation] if we use the popular

third-party corpus Jieba, but we do not want to see such a result when we add the field "analytic" to the user lexicon. When the "parser" field is added to the user's dictionary, the result will be [function,parser]. Deactivated words are also called "dummy words in computer search, non-search words". In search engines, in order to save space and search efficiency, certain words or phrases are usually automatically ignored in search requests, and these words or phrases are called deactivated words. The deactivation dictionary is a filter composed of a number of deactivation words, in the named entity recognition of the word, the system can be based on the deactivation dictionary to filter out some of the words or words that are not useful, to improve the accuracy of the word and the system computing efficiency. Deactivated words mainly include common pronouns, inflectional auxiliaries, adverbs, prepositions, conjunctions, etc., which usually have no obvious meaning of their own and only have a certain role when they are put into a complete sentence, such as: [you, I, he, this, that, the, in, then], etc.

- Tokenization: In the process of constructing mathematical knowledge graphs, word separation is mainly used to obtain mathematical knowledge points and related mathematical vocabulary, so there is no strict requirement on the lexicality of the words obtained by word separation. In this paper, we use Hanlp natural language processing toolkit to classify the text by perceptual machine. Hanlp has the features of perfect function, high performance, clear architecture, new corpus and customizable. and can recognize new words

2.5 Experiment

2.5.1 Dataset and Environment

The accuracy of the knowledge point extraction directly affects the construction of the feature database. The experiment selected 100,000 and 500 high school mathematics test questions with knowledge points in the web crawler as samples, and divided them into two parts for the automatic extraction of knowledge points without and with expert audit. As shown in Table , the accuracy of knowledge point extraction is 72.3% when there is no expert review and 96.6% when there is expert review. This experiment shows that the real-time updating of the knowledge map of high school mathematics in the process of automatic knowledge point extraction is the key to improve the accuracy of knowledge point extraction.

	Without expert review	With expert review
Sample Size	100000	500
Accuracy	72.3%	96.6%

表 2.1 The accuracy comparison table

2.5.2 Performance Indicators

2.5.3 Design of Experiment

2.5.4 Experiment Review

2.5.5 The Exercise Embedding

2.6 Summary

This chapter introduces the automatic knowledge extraction method of test questions based on knowledge graph. The intelligent algorithm based on individual features proposes to convert the standard question bank into a feature bank with knowledge feature representation, and defines each dimension in the feature bank, constructs the individual feature bank for users according to individual knowledge points, and explains the recommendation algorithm of test questions in the intelligent question bank; the automatic extraction method of knowledge points of test questions based on knowledge map introduces the whole process of knowledge map of high school mathematics from corpus collection, construction, storage and update. This chapter also introduces the automatic extraction process of knowledge points of high school mathematics test questions. This chapter also presents a comparative experiment on the objectivity of feature database construction and the effectiveness of automatic knowledge point extraction, and the related analysis of the experimental results.

Chapter 3

Graph Attention Networks Embedded Knowledge Tracing Model With Transformer

3.1 Motivation

This section is a core part of this recommendation system, which is to obtain the student's knowledge mastery status by means of knowledge tracing. knowledge tracing is to model students' knowledge mastery based on their past answer records to obtain students' knowledge status. knowledge tracing models are abundant, and early knowledge tracing models are generally based on the Bayesian knowledge tracing (BKT)[10] of first-order Markov models, which are based on the assumption that student answers are based on a series of knowledge points that are considered to be unrelated to each other and therefore represented independently of each other, an approach that cannot capture the relationships between different concepts nor characterize complex conceptual transformations. In 2015, Piech et al. proposed the Deep knowledge tracing Model (DKT)[6], the first application of a long short-term memory network (LSTM) to the knowledge tracing task, which does not require knowledge point annotation but contains an implicit state of knowledge, achieving a baseline performance over BKT at that time, and it marked the prologue of knowledge tracing research based on neural network models. However, DKT could not output the hidden state of knowledge, and the interpretability was insufficient. Moreover, DKT used to store all memories in a hidden vector, and the prediction performance was not satisfactory for

long sequences. To address this problem, Memory Augmented Neural Network (MANN)[8] was proposed, which allows the network to keep multiple hidden state vectors and read and write these vectors separately. In 2017, Zhang et al. proposed Dynamic Key-Value Memory Networks (DKVMN)[11], which refers to the design of MANN for knowledge tracing tasks and optimizes MANN for knowledge tracing tasks with different input and output domains. DKVMN uses key-value pairs as the memory structure, which can avoid over better prediction performance relative to BKT and DKT is achieved by using key-value pairs as the memory structure, which can avoid overfitting, fewer parameters, and automatic discovery of similar exercises through latent concepts. The model also has better interpretability, as it stores the problem-related potential concepts in the key matrix and the mastery of the concepts in the value matrix, and updates the value matrix by correlating the input exercises with the key matrix. However, these models still suffer from the problem of long dependencies. To solve this problem, Sequential Key-Value Memory Networks (SKVMN)[1], which designs a hop-LSTM structure to aggregate the hidden states of similar exercises, was proposed. However, existing models often do not consider enough the higher-order connections between knowledge points, or simply treat knowledge points as mutually independent nodes, or treat knowledge points as simple hierarchical models, but in fact, knowledge points are higher-order graph-like structures, in which case, using graph neural networks to characterize the relationships between knowledge points, training problem embedding and concept embedding is a better choice, so in this chapter, we propose a graph neural network-based knowledge tracing model, which uses higher-order problem-knowledge point relationships to solve the sparsity and complex knowledge point dependency problems, and at the same time, it can learn the mastery speed of students through the problem logging module, so as to better characterize the knowledge tracing process.

3.2 Related Theory

3.2.1 Knowledge Tracing

Knowledge tracing provides the conditions for intelligent and adaptive education, which is personalized and automated. Knowledge tracing tasks track changes in students' knowledge status from their historical learning records, predict future learning performance, and provide targeted learning coaching. The essence is to obtain the current learning status based on the student's past learning performance to predict the future learning performance. The actual

practice is to analyze the data and process modeling of students' past answer records, so as to model the current student learning status data, and let the model follow the learning status of students at each stage, and predict the probability of correct answers to the exercises based on the learning status data. In subject learning, subject knowledge consists of a series of knowledge points, and students' learning status is actually based on their mastery of each knowledge point. The general form of knowledge tracing is that given a sequence of student answers, which consists of a series of exercises associated with a specific knowledge point, a basic assumption in knowledge tracing is that student performance is based on mastery of the knowledge point. The student's performance can be used to infer the student's mastery of each knowledge point, i.e., the mastery of the learning state.

The mathematical representation of the knowledge tracing task is an interactive answer record $X_t = (x_1, x_2, , x_t)$ of a student on an exercise sequence, based on which the student's learning status is obtained by modeling and predicting the student's performance in the next exercise x_{t+1}. Where x_t is usually represented as an ordered pair (q_t, a_t), the ordered pair indicates that the student answered the question q_t at time t, and a_t indicates the score of the question, also many knowledge tracing tasks are represented by correct or incorrect answers, when a_t is 0 or 1. In this case, what is actually predicted is the probability of answering correctly for the next question $P(a_{t+1} = 1|a_t, a_{t-1}, ..., a_1)$. As shown in the figure **??**.

3.2.2 Graph Neural Networks

The traditional neural network model, which has achieved greater success in extracting features from Euclidean space data, has not worked well for applications on non-Euclidean spaces. And currently, many practical application scenarios are formally generated in non-Euclidean spaces. In the process of knowledge tracing, the connection between knowledge points is exactly a structure of non-Euclidean space, i.e., knowledge graph structure. The irregularity and complexity of graphs pose a great challenge to traditional deep learning algorithms, and the variability of the connections of the nodes of graphs leads to a huge computational effort, which makes some traditional neural network models that perform well in Euclidean space cannot be directly applied to non-Euclidean space. For example, in processing images where each pixel of the image is independent of each other, convolutional neural networks are easier to compute and parallelize for structures like images that are stable and single and have individual parts independent of each other, but this is not the case for graph structures where each node is connected to and has dependencies on other nodes. To

characterize the interdependencies between graph nodes, Graph Neural Networks (GNNs) are proposed.

There are five major classes of graph neural networks: Graph Convolution Networks (GCN), Graph Attention Networks(GAN), Graph Autoencoders(GAE), Graph Generative Networks(GGN), and Graph Spatial-temporal Networks(GSN). GCN, the pioneer of graph neural networks, simply applies the convolution operation in image processing to graph structured data processing and gives a specific derivation. It actually aggregates the features of graph neighbor nodes to make a linear transformation. To solve the problem of insufficient depth of graph propagation, multiple GCN layers can be stacked to capture the k-hop neighbor node information. However, the GCN processing requires putting the whole graph into the computational memory, which limits its computational performance. It also requires pre-inputting the graph structure information, which limits its application. To solve the problem that GCN aggregates neighbor nodes ignoring different weights of different neighbor nodes, Graph Attention Networks (GAT) introduces a masked self-attention mechanism to compute the representation of each node by assigning different weights according to the neighbor node characteristics. It does not need to use a pre-constructed graph, so only the neighbor nodes need to be known, which greatly increases the computational speed.

Graph embedding is an important research topic in graph neural networks, which represents nodes in a graph as low-dimensional vectors by preserving the network topology and node information of the graph, which is then processed by other machine learning algorithms. In this chapter, we characterize the association between knowledge points and knowledge point information by a graph embedding algorithm to obtain the learning of embedding for knowledge structures.

3.2.3 Transformer

3.3 Proposed Model

3.3.1 Algorithm Overview

The key point of this section is to track students' knowledge. This paper proposes a knowledge tracking model based on graph self attention network and Transformer model GATKT. In the experimental verification phase, the baseline performance of the model on several datasets achieves SOA. The first layer of this model is the embedding aggregation

layer. In this layer, a gat is used to aggregate the problem embedding and the knowledge point embedding, and to show the internal relationship between the problem and the knowledge point. It can solve the problem of data sparsity and the problem of knowledge point dependence and complex relationship. The second layer is a transformer based knowledge tracking model. Referring to the famous transformer model proposed by Google Brain [9], this layer designs a knowledge tracking model based on the transformer mechanism, which inputs questions and skills embedding, and outputs the prediction of the correct answer to the next question. Through this mechanism, the model can realize the learning of long range dependence. In addition, the model can output the hidden knowledge state through the decoder, which is the key to exercise recommendation in the next stage. In a word, the overall architecture diagram of the model can be seen in Figure 3.1, which is divided into the following modules:

1. GAT-based embedding layer: embedding layer based on gat: this layer uses embedding method to represent questions, knowledge points and answers. In the graph neural network, each problem represents a node, which is connected with several knowledge nodes, and these knowledge nodes are connected with the related problem nodes. Considering the relevance of knowledge points, knowledge points and knowledge points can be directly connected. The structure is shown in the figure 3.2. This embedding layer does not carry out pre training, but carries out overall training with other layers to optimize the final result.

2. GAT-based knowledge state tracking layer: this layer establishes a graph of knowledge points, which is used to track the students' mastery of knowledge points. This layer updates the knowledge state through the knowledge state, and finally it will be used in the subsequent recommendation process.

3. Knowledge tracing layer based on Transformer: this layer is similar to the traditional transformer structure, and has two parts: encoder and decoder. It learns the weight matrix of knowledge points and problems, and can solve the problem of knowledge point dependence. Through the attention mechanism, it can also capture similar problems of knowledge points.

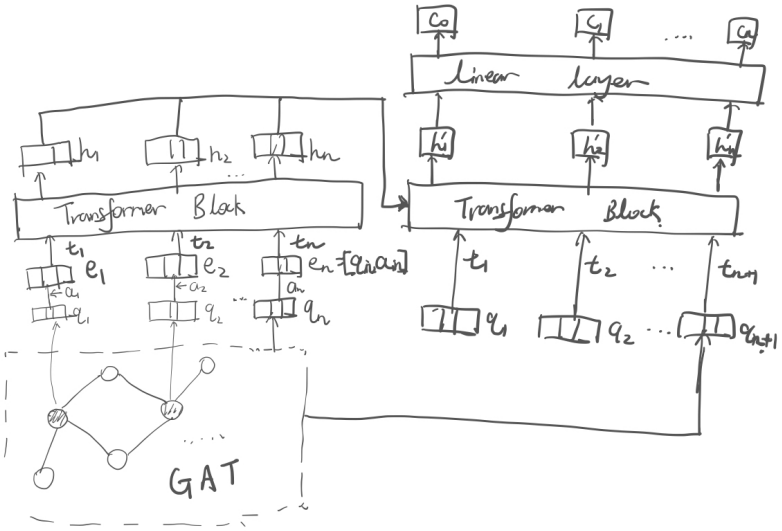

Fig. 3.1 Overview Model Structure

3.3.2 GAT-based Embedding Layer

In this section, the graph self attention network is used to embed the problem knowledge point association graph. The embedding method can reduce the sparsity in representing the knowledge points in question data. Let's denoted the i-th question as q_i. The answer to question I is a_i. The answer record is $x = \{x_1, x_2, ...x_{T-1}\}$, where $X_i = (q_i, a_i)$. Based on the answer records of the previous $t-1$ times, we predict the correct probability of the answer to question t. For each question q_i, there are a series of knowledge points related to it, these knowledge points are denoted as $\{p_1, p_2, ..., p_{n_i}\}$, n_i marks the number of knowledge points related to the question q_i. Similarly, a knowledge point p_j has also several related question $q_1, ..., q_{n_j}$, where n_j is the number of question related to the knowledge point p_j. So a graph $\mathscr{G}$ can be used to represent the relation of question-knowledge point, where $\mathscr{G} = \{(q, r_{qp}, p) | q \in \mathscr{Q}, s \in \mathscr{S}$ and $\mathscr{Q}$ denotes question sets and $\mathscr{P}$ denotes knowledge point sets respectively. $r_{qp} = 1$ if q is related to p.

In this part, embedding calculation can obtain the deep correlation between problems, that is, the correlation of knowledge points. This can be achieved with graph structure. Intuitively speaking, we can also infer the correct answer probability of the question by mastering the knowledge points. This method can also establish a better representation of

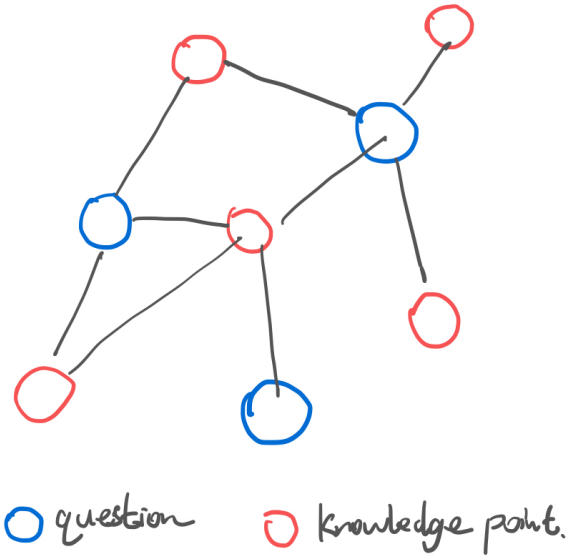

Fig. 3.2 The Graph of Knowledge Point and Question

problem knowledge points. In the question-knowledge graph, GAT can capture the high order relation for n-hop neighboring nodes. Also, GAT uses the idea of transformer for reference and introduces masked self attention mechanism. When calculating the representation of each node in the graph, it will assign different weights to its neighbors according to their different characteristics.

GAT applies an attention mechanism to weighted summation of neighboring node features. The weights of the neighboring node features depend entirely on the node features and are independent of the graph structure. In GAT, each node in the graph can be assigned different weights to neighboring nodes based on their characteristics. With the introduction of the attention mechanism, it is only relevant to adjacent nodes, i.e., nodes that share edges, without the need to get information about the whole graph. Specifically, the node of GAT network represents the knowledge point embedding or question embedding. Denote the node as i and the neighbor node set as $\mathcal{N}_i$ and the embedding of node as x_i, the input of the graph attention layer is $x = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{x_3} \ldots \overrightarrow{x_n} | \overrightarrow{x_i} \in \mathbb{R}_F\}$ and the output is $x = \{\overrightarrow{x_1}', \overrightarrow{x_2}', \overrightarrow{x_3}' \ldots \overrightarrow{x_n}' | \overrightarrow{x_i}' \in \mathbb{R}_{F'}\}$, where n is the number of nodes and F is the number of node features. represents the features of all nodes, and F' denotes the output node features. In order to get the corresponding input and output transformations, we need to perform at least

one linear transformation based on the input features to get the output features, so we need to train a weight matrix for all nodes: $W \in \mathbb{R}^{F' \times F}$, and this weight matrix is the relationship between the F features of the input and the F′ features of the output.

To achieve better high dimension feature representation, we need at least one linear transformation from low-dimensional to high-dimensional features. Therefore, a linear transformation is first done for the node features and then the coefficients are calculated. The attention mechanism of self-attention is implemented for each node, and the attention coefficients (Attention coefficients) and its softmax value are:

$$e_{ij} = a\left(\mathbf{W}\vec{x}_i, \mathbf{W}\vec{x}_j\right)$$

$$\alpha_{ij} = \text{softmax}_j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(e_{ik}\right)}$$

The e_{ij} and α_{ij} represents the importance of node i to node j.

In General, the complete attention mechanism is:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLu}\left(\overrightarrow{\text{a}}^T\left[W\vec{h}_i \| W\vec{h}_j\right]\right)\right)}{\sum_{k \in N_i} \exp\left(\text{LeakyReLu}\left(\overrightarrow{\text{a}}^T\left[W\vec{h}_i \| W\vec{h}_k\right]\right)\right)}$$

, where $||$ means concatenation operation. It is a single-layer feedforward neural network, $\overrightarrow{\text{a}} \in \mathbb{R}^{2F'}$ is the weight matrix connecting the layers in the neural network, and a LeakyReLu function is also added to the output layer of this feedforward neural network. The procedure is like Figure 3.3;

Here, since the graph structure information is to be considered, the concept of masked attention is introduced, which means that the denominator of the softmax considers only the first-order information about the neighbors. By this method, each node then gets a weight about the neighboring nodes and then the combined representation of the nodes is obtained by summing.

$$\vec{x}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{x}_j\right)$$

where σ are the non-linear transformation such as ReLU or LeakyReLU. If the edge $j \to i$ does not exist, we can simply omit the calculation of α_{ij}, which can reduce the calculation.

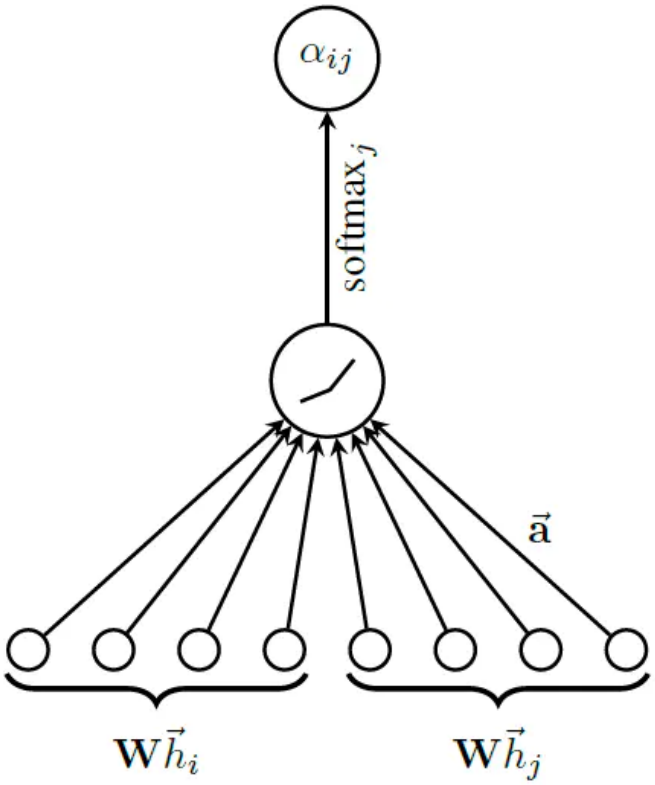

Fig. 3.3 Feature extraction and attention mechanism

The formula means that the output characteristics of this node are related to all the nodes adjacent to it and are obtained after the nonlinear activation of their linear sum.

To make the result of self-attention more stable. The multi-headed attention mechanism can be used here, like Figure 3.4. Multi-head attention is actually a combination of multiple self-attention structures, each head learns features in a different representation space, and multiple heads may learn slightly different attention focus, which gives the model more capacity. K independent attention mechanisms can be achieved by connecting k features together.

$$\vec{x}_i' = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{x}_j\right)$$

K-averaging is used to replace the join operation and to delay the application of the final nonlinear function. The attention coefficients between different nodes after regularization are obtained by the above operation and can be used to predict the output characteristics of each

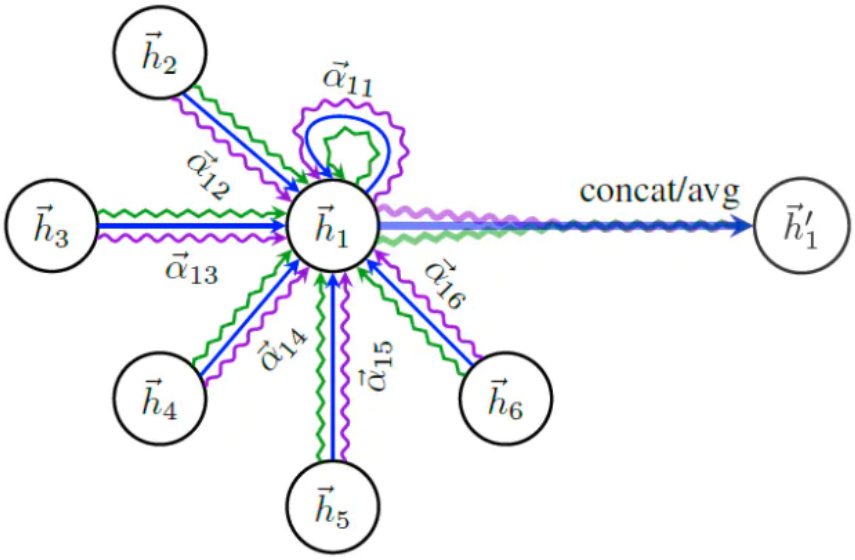

Fig. 3.4 Multi head Attention

node:

$$\vec{x}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k \mathbf{W}^k \vec{x}_j\right)$$

There are a total of K attention mechanisms to consider, with k denoting the kth of K.

From this layer, we got the embedded question and knowledge point vector $\hat{q}_i = \vec{x}'_i$, which would be passed into the Transformer Block in Knowledge tracing module.

3.3.3 Transformer-based knowledge tracing

The input of transformer Block is the concatenation of the embedded question $\hat{q}_i$ and corresponding answer a_i, i.e. $e_i = \hat{q}_i \| a_i$. The general structure is like Figure 3.5 and the Transformer-block is like Figure 3.6. The traditional Transformer model is modified to apply to knowledge tracing task. The Transformer can learn the W-matrix relating underlying knowledge points and questions rather than directly learn the representation of each question. This allows the model to learn deeper connections between problems, resulting in better inference performance. It can cluster problems with similar knowledge points and reduce the variance for less frequent problems.

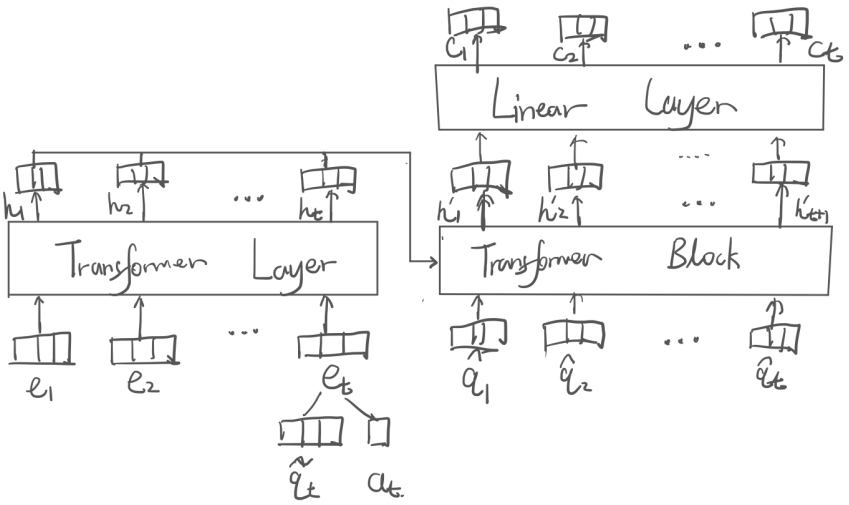

Fig. 3.5 The Transformer architecture

The output of Embedding layer e_i is passed into a Transformer Block, which is a self-attention module.

$$q_i = Qe_i, k_i = Ke_i, v_i = Ve_i$$
$$A_{ij} = \frac{q_i k_j + b(\Delta t_{i-j})}{\sqrt{d_k}}, \forall j \leq i$$
$$h_i = \sum_{j \leq i} \text{softmax}\left(A_{ij}\right) v_j$$

the masked attention layer first extracts query gi, key k;, and value y; from the inputs ei. It then assign an attention, Aii, to a past interaction e, based on two components:

1. $q_i k_j$, the query-key agreement between e_i and e_j, which could be interpreted as the degree of latent skills overlapping between interaction e_i and e_j;

2. A time gap bias, $b(\delta t_{i-j})$, which adjusts the attention weight by the time gap between interactions e_i and e_j. The hidden representation h_i is a weighted sum of the past value representations of e_i.

A Transformer block also have feedforward layer, normalization layer, and residual connections. The outputs of a stack of Transformer blocks is feed to the linear layer before calculating the final loss. e_{i+1} and e^*_{i+1} are the results of applying Interaction Mapping Layer to $(\hat{q}_{i+1}, 1)$ and $(\hat{q}_{i+1}, 0)$.

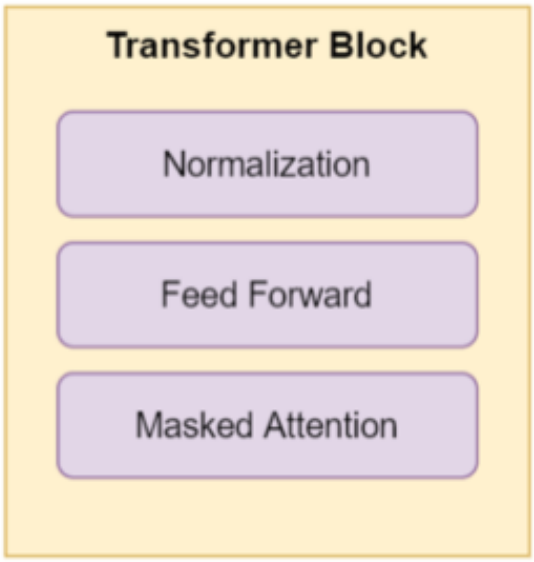

Fig. 3.6 The Transformer-block architecture

$$p_{i+1} = \frac{\exp(h_i e_{i+1})}{\exp(h_i e_{i+1}) + \exp\left(h_i e^*_{i+1}\right)}$$
$$\text{Loss} = -\sum_i c_{i+1} \log\left(p_{i+1}\right) + \left(1 - c_{i+1}\right) \log\left(1 - p_{i+1}\right)$$

3.3.4 The Knowledge State

TODO

3.4 Experiments

In this section, we conduct several experiments to investigate the performance of our model. We first evaluate the prediction error by comparing our model with other baselines on three public datasets. Then the paper make ablation studies on the GAT and GATKT to show their effectivenes. Finally, the paper evaluates the design decisions of the recap module to investigate which design performs better.

3.4.1 Datasets

To evaluate our model, the experiments are conducted on three widely-used datasets in KT and the detailed statistics are shown in Table 3.1.

- ASSIST09 was collected during the school year 2009-2010 from ASSIST ments online education platform. We conduct our experiments on" skill- builder " dataset. Following the previous work 32, we remove the duplicated records and scaffolding problems from the original dataset. This dataset has 3852 students with 123 skills, 17, 737 questions and 282,619 exercises

- ASSIST12 was collected from the same platform as ASSIST09 during the school year 2012-2013. In this dataset, each question is only related to one skill, but one skill still corresponds to several questions. After the same data processing as ASSIST09, it has 2, 709, 436 exercises with 27, 485 students, 265 skills and 53, 065 questions.

- Ednet was collected by 51. As the whole dataset is too large, we randomly select 5000 students with 189 skills, 12, 161 questions and 676, 974 exercises

表 3.1 Dataset Statistics

	ASSIST09	ASSIST12	EdNet
students	3,852	27,485	5000
questions	17,737	53,065	12,161
skills	123	265	189
exercises	282,619	2,709,436	676,974
questions per skill	173	200	147
skills per question	1.197	1.000	2.280
attempts per question	16	51	56
attempts per skill	2,743	10,224	8,420

Note that for each dataset we only use the sequences of which the length is longer than 3 in the experiments as the too short sequences are meaningless. For each dataset, we split 80% of all the sequences as the training set, 20% as the test set. To evaluate the results on these datasets, we use the area under the curve (AUC)as the evaluation metric.

3.4.2 Baselines

In order to evaluate the effeciveness of our proposed model, we use the following models as our baselines.

- BKT uses Bayesian inference for prediction, which models the knowledge state of the skill as a binary variable.

- DKT is the first method that uses deep learning to model knowledge tracing task. It uses recurrent neural network to model knowledge state of students
- DKVMN uses memory network to store knowledge state of different concepts respectively instead of using a single hidden state
- DKT-Q is a variant of DKT that we change the input of DKT from skills to questions so that the DKT model directly uses question information for prediction
- DKIT-QS is a variant of DKT that we change the input of DKT to the concatenation of questions and skills so that the DKT model uses question and skill information simultaneously for prediction

3.4.3 Performance

Table 3.2 reports the AUC results of all the compared methods. From the re sults we observe that our GATKT model achieves the highest performance over three datasets, which verifies the effectiveness of our model. To be specific, our proposed model GATKT achieves at least 1% higher results than other baselines Among the baseline models, traditional machine learning models like BKT and KTM perform worse than deep learning models, which shows the effectiveness of deep learning methods. DKVMN performs slightly worse than DKT on average as building states for each concept may lose the relation information between concepts. On the other hand, we find that directly using questions as input may achieve superior performance than using skills. For the question-level model DKT-Q, it has comparable or better performance than DKT over ASSIST12 and Ednet datasets. However, DKT-Q performs worse than DKT in ASSIST09 dataset. The reason may be that the average number of attempts per question in ASSIS'T09 dataset is significantly less than other two datasets as observed in Table 1, which illustrates DKT-Q suffers from data sparsity problem. Besides, the AUC results of the model DKT-QS are higher than DKT-Q and DKT, except on ASSIST12 as it is a single-skill dataset, which indicates that considering question and skill information together improves overall performance.

3.5 Summary

In this chapter, we propose a knowledge tracing model to employ high-order question-skill relation graphs into question and skill representations. Besides, to model the student's

表 3.2 AUC results over three datasets

Model	ASSIST09	ASSIST12	EdNet
BKT	0.6571	0.6204	0.6027
KTM	0.7169	0.6788	0.6888
DKVMN	0.7550	0.7283	0.6967
DKT	0.7561	0.7286	0.6822
DKT-Q	0.7328	0.7621	0.7285
DKT-QS	0.7715	0.7582	0.7428
GATKT	**0.7896***	**0.7754***	**0.7523***

mastery for the question and related skills, we design a recap module to select relevant history states to represent student's ability. Then we extend a generalized interaction module to represent the student's mastery degree of the new question and related skills in a consistent way.

Chapter 4

Exercise Recommendation System Based on Knowledge State

4.1 Motivation

High school mathematics intelligent question bank is built by learning from the model of adaptive learning. Adaptive learning is the theoretical abstraction of adaptive learning system model. Many adaptive learning system models focus on the construction of domain model, student model and adaptive engine. This paper proposes an intelligent recommendation algorithm based on individual characteristics, which integrates domain model, student model and adaptive engine The self-adaptive engine abstracts the high school mathematics question bank into the characteristic question bank according to the domain model, and abstracts the student model into the individual characteristic bank. The self-adaptive engine is based on the individual characteristic bank through the intelligent algorithm, dynamically selects the training questions suitable for the individual from the characteristic question bank, and realizes the personalized recommendation of the question bank.

Traditional paper teaching materials are usually printed in large quantities. Due to the limitation of paper space and in order to adapt to the majority of students, one or two representative test questions have to be selected according to each chapter. In order to meet the comprehensiveness of knowledge points and the public adaptability, such teaching aids lead to the small scale of test questions, the incomplete coverage of questions and the failure to meet the requirements of students' targeted training. E-learning has brought a reform in the field of teaching with its flexible learning style, avoiding large class teaching, and

students can choose their own learning resources through autonomous navigation. Online question bank with eleaning As the times require, the early online question bank simply moves the offline resources to the Internet. Although students can choose their own topics and find the test questions they want for intensive training, it is lack of interactivity. Students do not get timely and effective feedback after finishing the questions, and can not accurately locate the strengths and weaknesses of students, which is very limited in improving students' performance. In view of these shortcomings, it is the current research direction of intelligent question bank to build a student-centered system, in which students can feed back their mastery of knowledge points in real time by doing questions, and build a complete user portrait of students. The question bank can intelligently diagnose, screen and push targeted questions according to the submission of students' questions.

Adaptive learning means that learners choose different learning styles depending on the content they are learning, and that they continue to discover their own "personalized" learning style as they learn. Adaptive learning is not a simple transfer of offline resources to online, nor is it a simple combination of "Internet + education". Rather, it focuses on the individual student and provides different personalized learning programs for people with different cognitive levels and styles.

The concept of adaptive learning has been a hot topic of research in the education field and has been one of the theories studied by many educators. Since Brusilovsky proposed a general model of adaptive learning system, the concept of adaptive learning was first applied to engineering practice, but there has been a lack of standard reference models and architectural systems until the first adaptive hypermedia application model was developed, and later there were LAOS, XML adaptive media model and WebXML reference model. Since then, adaptive learning has been transformed from theoretical research in education to application development in computing, and many adaptive learning models have focused on the construction of domain models, student models, and adaptive engines.

The domain model focuses on building student learning resources, which are not just student textbooks or tutorials, but also structured entities in the domain knowledge, and ontologies or semantic webs are used to build connections between entities to form a complete knowledge base, including fine-grained knowledge concepts and their attributes, examples, exercises, materials, and even graphical images and audio.

The student model is mainly used to build a model with individual student characteristics, which can be described as individual students' mastery of each test topic in the domain

model, and the student model will change with the changes of entity relationships in the domain model. In addition, the student model is also described in CELTS-11 according to our educational technology specification on the learner specification model, which adds additional information such as the student's age, background, and region.

The adaptive engine, also known as the instructional selection strategy, uses adaptive interpretation rules from the domain model to the student model to enable the system to select the domain model that meets the student's requirements based on the student's personality characteristics, either through rules or algorithms. Currently, it is popular to use big data technology to recommend students' personalized learning solutions by using popular recommendation algorithms such as collaborative filtering recommendation, content recommendation, and hybrid recommendation through statistics of students' learning paths or habits.

In this paper, we further optimize the domain model, student model, and teaching strategy in the adaptive reference model, and apply the optimized model to build an intelligent question bank system. The feature database is structured according to the dimensions of forgetfulness, abstractness, indirectness, comprehensiveness, and computational complexity, and individual students are trained to generate a database of individual features reflecting their knowledge acquisition ability in real time. The intelligent question selection algorithm is used to dynamically select the appropriate training set from the question database based on the individual feature database.

4.2 Proposed Model

4.2.1 Algorithm Overview

This chapter proposes an adaptive recommendation method for test questions that combines graph neural network knowledge tracking and collaborative filtering. The method models students' knowledge states through the GAT-based knowledge tracking model model designed in the previous chapter, and uses the clustering method of graph neural networks to perform knowledge-based clustering of test questions so that a series of relevant exercises can be obtained. The first part is the recall part, i.e., candidate test generation, which generates a series of similar exercises, and this is done by clustering the test questions through graph neural networks, which apply graph clustering methods to cluster the test questions based on knowledge points in a reasonable way. The second part is the ranking part, which

combines the knowledge state of students obtained from the knowledge tracking system with co-filtering.

1. Exercise candidate generation: The role of this part is to generate a series of exercises that match the student's current learning state, and quickly filter out a preliminary set of exercises, and the subsequent part will generate the most suitable N exercises from this set and sort them by collaborative filtering algorithms. It designs a multi-layer MLP network with reference to the youtube recommendation system. It concatenates the student learning state embedding vector generated by knowledge tracking with the student's problem record embedding, personalized information, etc. as input. After a series of ReLU networks and then output a probability distribution over all candidate exercises by softmax normalization.

2. Ranking: This part is to generate Top N recommendation exercise. It implements the ranking of the recommended test questions using a neural network architecture similar to that of the previous section. Its inputs are the user's knowledge structure, recent exercises, and common practice questions from similar users.

The structure of the recommendation system is like Figure. 4.1

4.2.2 Recall Stage

In the recall phase, due to the large amount of data, the algorithm of Collaborative Filtering (CF) can achieve half the result with twice the effort. It only needs to calculate the similarity of students' knowledge states h' with other students who have similar knowledge states h.

Collaborative Filtering, the most classic type of recommendation algorithm, includes both online collaborative and offline filtering. The so-called online collaborative is to find items that users may like through online data, while offline filtering is to filter out data that are not worth recommending, such as data with low recommendation value ratings, or data that users have already purchased despite high recommendation values. The model of collaborative filtering is generally m recommended items, m users' data, only some users and some data are rated between the data, other parts of the rating is blank, at this time we want to use the existing part of the sparse data to predict the rating relationship between those blank items and data to find the highest rated items recommended to users.

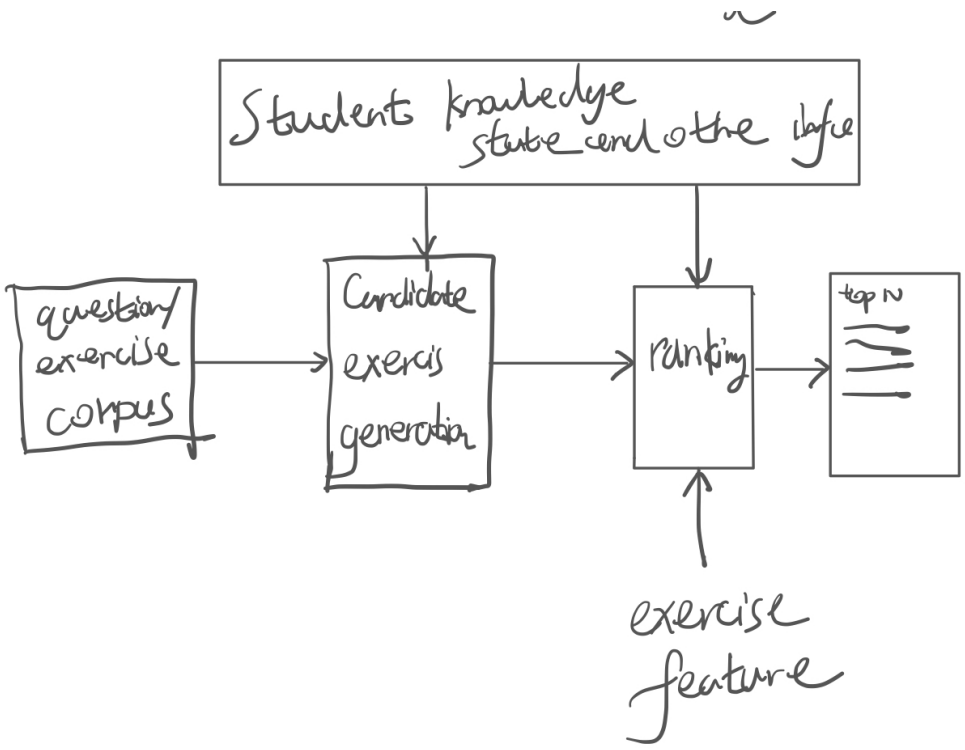

Fig. 4.1 The Architecture of Recommendation System

Generally speaking, there are three types of collaborative filtering recommendations. The first one is user-based collaborative filtering, the second one is item-based collaborative filtering, and the third one is model-based collaborative filtering.

Predict the similarity between users

User-based collaborative filtering mainly considers the similarity between users and users. As long as we find out the items that similar users like and predict the target users' ratings of the corresponding items, we can find the items with the highest ratings and recommend them to users. The item-based collaborative filtering is similar to the user-based collaborative filtering, except that we turn to find the similarity between items and users, and only if we find the ratings of certain items by target users, then we can predict similar items with high similarity and recommend the highest rated similar items to users. For example, if you buy a machine learning related book online, the website will immediately recommend a bunch of machine learning, big data related books to you, and the idea of collaborative filtering based on items is obviously used here.

Similar statistics are used to get neighboring users with similar hobbies or interests, so we can use the obtained user's knowledge state embedding h_t from the previous knowledge tracking module, which can be used as the basis for calculating the similarity. The first step is to find other users who are similar to the new user based on their historical behavior information; at the same time, to predict the items that the current new user may like based on the evaluation information of these similar users on other items. Given the user rating data matrix R, the user-based collaborative filtering algorithm needs to define the similarity function $s : U \times U \rightarrow R$ to calculate the similarity between users, and then calculate the recommendation results based on the rating data and the similarity matrix. We can use the cosine similarity to calculate this value.

$$s(u,v) = \frac{h_u \cdot h_v}{\|h_u\|_2 \|h_v\|_2}$$

where h_u and h_v represent the knowledge mastery state of user u and v. We can obtain the exercise recommendation history H of user B, which is closest to user A to be recommended, and then use the exercises in H as a rough set as the next sorted list of exercises.

Calculate the similarity between exercise

4.2.3 Ranking Stage

In the previous step we obtained a list of recommended exercises for similar students by collaborative filtering in the recall phase, and in this phase, a ranking of the exercises is needed to further reduce the amount of recommendations.

TODO:设计一个图神经网络用于知识点权重embedding or 简单的PMF等认知诊断算法

4.3 Experiment

4.3.1 Database

4.3.2 Baseline

4.3.3 Result

4.4 Summary

Chapter 5

Conclusion and Future Work

In this paper, a test question recommendation system based on graph neural network combined with knowledge mapping and factorization algorithm is established, which generally accomplishes the design goals and is original in the three stages of the recommendation system-data source generation, knowledge tracking and knowledge recommendation, etc. This paper adopts the current more popular graph neural network model and makes some improvements to the existing model to solve the tasks of knowledge analysis of test questions and knowledge state tracking of students, and achieves the desired results in the experiments with some performance improvements to the existing model.

Among them, in the data source part, they are input to the initial database by crawlers and manual input, and then the automatic knowledge point analysis model of test questions based on text mining and graph neural network iterative learning is used to complete the knowledge point label mining of test questions, and they then establish the knowledge map of high school mathematics by some means of knowledge map construction, which solves the "zero resource " problem. In terms of the knowledge tracking of students, this paper innovatively applies the graph self-attentive network to this task, and through the improvement of the model and the consideration of several factors, the task has been improved somewhat compared with the existing model and has improved in performance. Finally, by labeling the knowledge of the test questions and tracking the students' knowledge, the factorization machine algorithm is used to complete the learning resource recommendation, which solves the cold start problem and accomplishes the design goal of adaptive learning.

In future work, the sequential nature of the model can be solved by combining other graph neural network models or adding some memory mechanisms.

References

[1] Abdelrahman, G. and Wang, Q. (2019). Knowledge tracing with sequential key-value memory networks. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[2] Chaudhry, R., Singh, H., Dogga, P., and Saini, S. K. (2018). Modeling hint-taking behavior and knowledge state of students with multi-task learning. *International Educational Data Mining Society*.

[3] Cui, W., Xue, Z., and Thai, K.-P. (2018). Performance comparison of an ai-based adaptive learning system in china. In *2018 Chinese Automation Congress (CAC)*, pages 3170–3175. IEEE.

[4] Guo, H., Tang, R., Ye, Y., Li, Z., and He, X. (2017). Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*.

[5] Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J. (2016). Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 43–50.

[6] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., and Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Advances in neural information processing systems*, 28:505–513.

[7] Rendle, S. (2010). Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE.

[8] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

[10] Yudelson, M. V., Koedinger, K. R., and Gordon, G. J. (2013). Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer.

[11] Zhang, J., Shi, X., King, I., and Yeung, D.-Y. (2017). Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774.

附录 A

How to Install LaTeX

Windows OS

TeXLive package - full version

1. Download the TeXLive ISO (2.2GB) from
   https://www.tug.org/texlive/
2. Download WinCDEmu (if you don't have a virtual drive) from
   http://wincdemu.sysprogs.org/download/
3. To install Windows CD Emulator follow the instructions at
   http://wincdemu.sysprogs.org/tutorials/install/
4. Right click the iso and mount it using the WinCDEmu as shown in
   http://wincdemu.sysprogs.org/tutorials/mount/
5. Open your virtual drive and run setup.pl

or

Basic MikTeX - TeX distribution

1. Download Basic-MiKTeX(32bit or 64bit) from
   http://miktex.org/download
2. Run the installer

3. To add a new package go to Start » All Programs » MikTex » Maintenance (Admin) and choose Package Manager

4. Select or search for packages to install

TexStudio - TeX editor

1. Download TexStudio from
   http://texstudio.sourceforge.net/#downloads

2. Run the installer

Mac OS X

MacTeX - TeX distribution

1. Download the file from
   https://www.tug.org/mactex/

2. Extract and double click to run the installer. It does the entire configuration, sit back and relax.

TexStudio - TeX editor

1. Download TexStudio from
   http://texstudio.sourceforge.net/#downloads

2. Extract and Start

Unix/Linux

TeXLive - TeX distribution

Getting the distribution:

1. TexLive can be downloaded from
   http://www.tug.org/texlive/acquire-netinstall.html.

2. TexLive is provided by most operating system you can use (rpm,apt-get or yum) to get TexLive distributions

Installation

1. Mount the ISO file in the mnt directory

```
mount -t iso9660 -o ro,loop,noauto /your/texlive####.iso /mnt
```

2. Install wget on your OS (use rpm, apt-get or yum install)

3. Run the installer script install-tl.

```
cd /your/download/directory
./install-tl
```

4. Enter command 'i' for installation

5. Post-Installation configuration:
   http://www.tug.org/texlive/doc/texlive-en/texlive-en.html#x1-320003.4.1

6. Set the path for the directory of TexLive binaries in your .bashrc file

For 32bit OS

For Bourne-compatible shells such as bash, and using Intel x86 GNU/Linux and a default directory setup as an example, the file to edit might be

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/i386-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

For 64bit OS

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/x86_64-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

Fedora/RedHat/CentOS:

```
sudo yum install texlive
sudo yum install psutils
```

SUSE:

```
sudo zypper install texlive
```

Debian/Ubuntu:

```
sudo apt-get install texlive texlive-latex-extra
sudo apt-get install psutils
```

附录 B

Installing the CUED Class File

LaTeX.cls files can be accessed system-wide when they are placed in the <texmf>/tex/latex directory, where <texmf> is the root directory of the user's TeXinstallation. On systems that have a local texmf tree (<texmflocal>), which may be named "texmf-local" or "localtexmf", it may be advisable to install packages in <texmflocal>, rather than <texmf> as the contents of the former, unlike that of the latter, are preserved after the LaTeXsystem is reinstalled and/or upgraded.

It is recommended that the user create a subdirectory <texmf>/tex/latex/CUED for all CUED related LaTeXclass and package files. On some LaTeXsystems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For TeXLive systems this is accomplished via executing "texhash" as root. MIKTeXusers can run "initexmf -u" to accomplish the same thing.

Users not willing or able to install the files system-wide can install them in their personal directories, but will then have to provide the path (full or relative) in addition to the filename when referring to them in LaTeX.