Research on Knowledge Discovery and Affective Computing for Short Text Processing

韓　朝

A Thesis submitted to Tokushima University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

March 2019

Tokushima University
Graduate School of Engineering
Information Science and Systems Engineering

Contents

Acknowledgment

First, I would like to thank my adviser Professor Fuji Ren and all the teachers of Tokushima University. With their help, I successfully completed my Ph.D. research. In the process of scientific research, from the selection of research interests to the publication of the research articles, and also the writing of the graduation thesis, my teachers gave me careful guidance and taught me a lot of scientific methods. And at the same time, they also help me understand Japanese vocabulary and culture, and have given me great care during my study in Japan.

I would like to thank Professor Duanqian Miao, the adviser of my Chinese university. Thank him and Professor Ren for giving me the opportunity to participate in the double-degree cooperation project. This project made me know more new teachers and friends and learn more different knowledge and culture.

I would like to thank all the students in Ren lab. I have learned a lot from them, from both Chinese friends and Japanese friends.

I would like to thank Mr. Xin Kang and Ms. Asada, who gave me a lot of guidance during the graduation procedure and thank them for helping me to submit the graduation materials on time.

Finally, I would like to thank all the teachers of the Thesis Examination Committee, Prof. Kenji Terada and Prof. Masami Shishibori. Thank you for taking time to read my thesis in your busy schedule, and sincerely thank you for your valuable advice and suggestions.

Abstract

Short text is often used in the QA system, and the processing method of short text has an important influence on the performance of the QA system. This thesis deals with short text information from three aspects: semantic, knowledge and emotion. Details are:

[1] Because of the uncertainty of both the language representation and the knowledge representation, the current methods for short text processing are not very effective. To solve the uncertainty of knowledge representation, a rough set knowledge discovery method for Chinese short text QA system is proposed. It uses the method of rough set equivalence partitioning to represent the rough set knowledge of the QA pairs, then uses the idea of attribute reduction to mine out the upper approximation representations of all the knowledge items. Based on the rough set QA knowledgebase, the knowledge match value of a QA pair can be calculated as a kind of knowledge item similarity. After all the knowledge similarities of one question and its answer candidates are given, the final matching values which combine rough set knowledge similarity with traditional sentence similarity can be used to rank the answer candidates. The experiment shows that the proposed method can improve the MAP and MRR compared with the baseline information retrieval methods.

[2] A novel method for Short Text Information Retrieval based Chinese Question Answering is proposed. It is developed from the Discernibility Matrix based Rules Acquisition method. Based on the acquired rules, the matching patterns of the training QA pairs can be represented by the reduced attribute words, and the words can also be represented by the QA patterns. Then the attribute words in the test QA pairs can be used to calculate the matching scores. The experimental results show that the proposed representation method of QA patterns has good flexibility to deal with the uncertainty caused by the Chinese word segmentation, and the proposed method has good performance at both MAP and MRR on the test data.

[3] The accurate extraction of knowledge subject is not only one of the important processes affecting the matching accuracy of the QA system based on the knowledge base, but also one of the important processes of knowledge subject positioning based on the knowledge map. A sequence labeling method for knowledge subject analysis for short text

KBQA is proposed. From the perspective of rough set model and rough set attribute importance, combined with the existing named entity recognition and syntactic subject analysis, the sequence labeling method is used to optimize the sequence labeling results for knowledge subject extraction, and thus improve knowledge subject analysis ability of the overall system. The experiment results verify the validity of the method.

[4] In short text KBQA system, the performance of knowledge predicate analysis can affect the overall matching result of knowledge triple. The knowledge predicate analysis of Chinese short text question is difficult because of the uncertainty of Chinese knowledge predicate representation. Based on the rough set theory, a new definition of knowledge predicate analysis of KBQA system was given, and a new method was proposed to analyze the knowledge predicate of the question. It can reduce the words which are weakly related with the knowledge predicate, and then the words which are more related with knowledge predicate representation will be used to match the knowledge triples to improve the overall performance of the system. The experiment results verify the validity of the method.

[5] A sentence-level sentiment analysis method is proposed to deal with sentiment measurement and classification problems. It is developed from a model called Synthetic and Computational Language Model(SCLM), which represents modifying and modified information respectively using matrices and vectors. In the proposed method, a global modifying matrix of a sentence is constructed and determinant value of this matrix is calculated and adjusted, and then the final value is used as the sentiment value of the sentence. The regression experiment shows that the deviation between the output sentiment and target sentiment does not exceed a class distance of 5-classes. The classification experiment shows that the proposed method has improved most of the performance compared to the simplified SCLM.

Chapter 1

Introduction

Short text applications are mainly concentrated in the fields of instant messaging, social networking service, and question answering system(QA system). Among them, QA system is a hot topic in the field of NLP research. In recent years, research and development related to dialogue systems have received increasing attention from major IT companies and research institutions. In various fields such as customer service, medical assistance, and life entertainment, some companies and organizations have begun to try to put the corresponding products into application. Based on huge data storage capabilities and fast computing power, these products are often able to perform tasks or services better than humans, or provide diverse entertainment content for humans. Compared to long texts used in news or documents, short texts have more flexible language expressions and more uncertainty, so there are many difficulties in dealing with them.

In the dialogue system or QA system, the uncertainty and handling difficulties of short texts are mainly reflected in three aspects, semantic, knowledge, and sentiment. Semantics is the basic element that constitutes a question and a sentence. The QA knowledge is the semantic and logical relationship between the question and the answer. The emotion is a supplement to the dialogue semantics. Therefore, in this thesis the study of short texts in QA system starts from these three perspectives.

The content of this thesis is arranged as follows: chapter 1 (this chapter) is the introduction of the full theis and will introduce some background and recent related works, chapter 2 and chapter 3 is about semantic and knowledge processing of Chinese short text of Document based Question Answering System (DBQA), chapter 4 and chapter 5

is about semantic and knowledge processing of Chinese short text of Knowledge based Question Answering System(KBQA), chapter 6 is about sentiment analysis of short text, and chapter 7 is the conclusion and future work.

1.1 Background

QA system is a system in which human users interact with computers through natural language as the ultimate goal, and the earliest well-known QA system was considered to be the Eliza QA robot [1] developed by MIT in 1966. Eliza can extract keywords from the text input by the user, and trigger the associated alternative reply statement according to the keywords and return the alternative sentence as answers[2]. Eliza's initial research goal was to use psychosocial therapy, after the advent of Eliza, researchers began experimenting with the extension of QA system technology to a wider range of medical or hospital-assisted areas during the 1970s and 1980s. For example, HHospital QA system can provide a simple initial health inquiry service, or a case inquiry service for doctors, or a corresponding dispensing program for some symptoms [3].

Later, when some researchers used the text QA system as the research object, another part of the researchers began to study the auxiliary procedures that can assist the deaf or person with physical impairment. In the late 1980s, the program framework of some research results began to consider components such as speech recognition, knowledge base construction, information retrieval(IR) , text generation, etc. It can be treated as a prototype framework for the speech QA system.

After the 1990s, the QA system began to gradually transition from an information retrieval framework to a complex system framework based on semantic understanding, knowledge understanding and model-based learning in order to accommodate higher application requirements. For example, the QA system introduced in the reference [4] greatly improves the matching accuracy of the results obtained by question retrieval by using the semantic knowledge base Wordnet[5]. In addition, machine learning (Machine Learning) and Neural Networking (Neural Networking, NN) has also begun to be used in research to improve the performance of QA systems [6, 7].

After entering the 20th century, Dialogue Management technologies has begun to re-

ceive attention [8]. And also, with the support of Internet technology, the related research on the open domain QA system has been launched [9]. In this period, statistical models have also been used in the QA system research [10]. The evaluation criteria and indicators of QA system are also gradually formed [11].

In recent years, along with Natural Language Processing (Natural Language Processing, NLP)[12], Big Data Mining (Big Data) Mining)[13], cloud computing (Cloud Computing) [14] and deep learning (Deep Learning) [15] and other major technological, the QA system technology has also made corresponding breakthroughs. First of all, the development of the corresponding theory and technology solves a large number of NLP problems in QA systems, and also provides the original data source for system application development and knowledge base construction. Secondly, the performance improvement of hardware and algorithms makes massive data processing become possible, with the information retrieval technology, the limitations of the knowledge domain of a QA system have been broken. The QA system is no longer limited to the previous single-inquiry service application, but has been developed for multi-domain, large-scale data or knowledge understanding ability.

QA system framework is mainly based on semantic analysis, affective computing, context understanding, dialogue management, topic extraction, hierarchical or granular classification or clustering, information retrieval, and natural language generation, etc.. Many famous toolkits such as Stanford CoreNLP toolkit [16] and OpenNLP toolkit [17] have been developed to deal with basic NLP task such as sentence breaks, vocabulary or fixed phrase segmentation, stem extraction, POS (the Part-of-Speech) annotation, named entity recognition(NER), syntax and dependency analysis and simple sentiment analysis.

After obtaining the basic language information based on the existing tools, the model and the algorithm can be used to obtain the information of different granularities contained in the language, including semantics, topics, opinions, emotions and so on. The system then needs to combine the knowledge base and database information to retrieve the information that needs to be returned. The information that needs to be returned can be the result directly returned to the user, or it can be the part contained in the information that returns the result. Therefore QA system is inseparable from the information retrieval module and the knowledge database. There has been a lot of open source

information retrieval framework [18], such as Lucene[19], Indri[20], etc., and also the open source knowledge database such like Freebase[21] and DBpedia[22] .

In addition to the corresponding theoretical and technical support, the training and construction of the QA system require a large amount of corpus data support. At present, the open source corpus resources for QA system are relatively scarce. Most of the research is based on the corpus generated in the interaction process of short text social networks such as Weibo and Twitter. UbuntuChatRoom Corpus[23] and MPC corpus[24] are two opensource English QA corpus. The content of Ubuntu ChatRoom Corpus is mainly generated by the Ubuntu system users, and the content is about the multi-person discussion on the existing problems during using Ubuntu. The content of the MPC corpus mainly comes from social networking, so its topic category is wider than the UbuntuChatRoom corpus. There are also QA corpora published in international conferences or evaluation competitions, such as TREC QA Track[25] and NTCIR QALab[26]. In China, the NLPCC has started the open domain Chinese QA system evaluation competition since 2015[27] . SMP has started the human-machine dialogue system evaluation competition since 2017[28] .

Many research institutions have open-sourced QA system framework based on IR technology such as AliceRobot[29], OpenEphyra[30], ChatterBot[31] . AliceRobot is based on the development of Eliza, the main principle is still simple based on retrieval and result matching; OpenEphyra is relatively complex compared to Alice and Eliza, which combines NLP, knowledge base retrieval and other modules; ChatterBot is a framework developed by Python language and can be combined with Python's extensive toolkit. However, there is still a lack of open source framework based on natural language generation technology.

From a technical point of view, the QA system is mainly divided into two categories: one is Document Based Question Answering System(DBQA System), also known as the Text QA System; the other is Knowledge based Question Answering System(KBQA)[32]. DBQA, that is, if a candidate answer sentence has been given, one or more candidate sentences for the best matching question are returned for the input question. Figure 1.1 shows an example of DBQA:

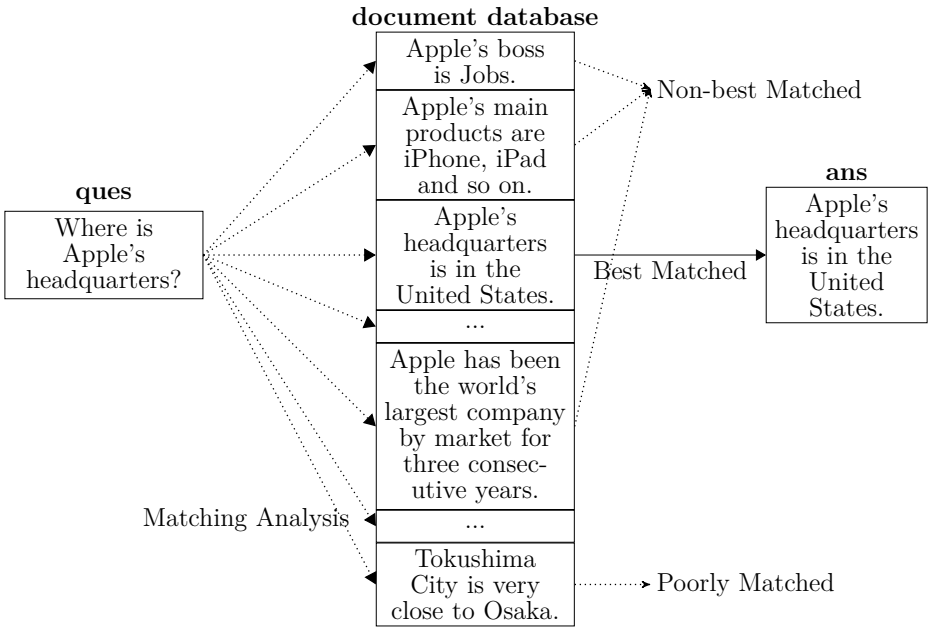

Figure 1.1: document based question answering system

KBQA, that is, there are no candidate answers, directly constructs the answer and returns based on the knowledgebase. Figure 1.2 shows an example of KBQA.

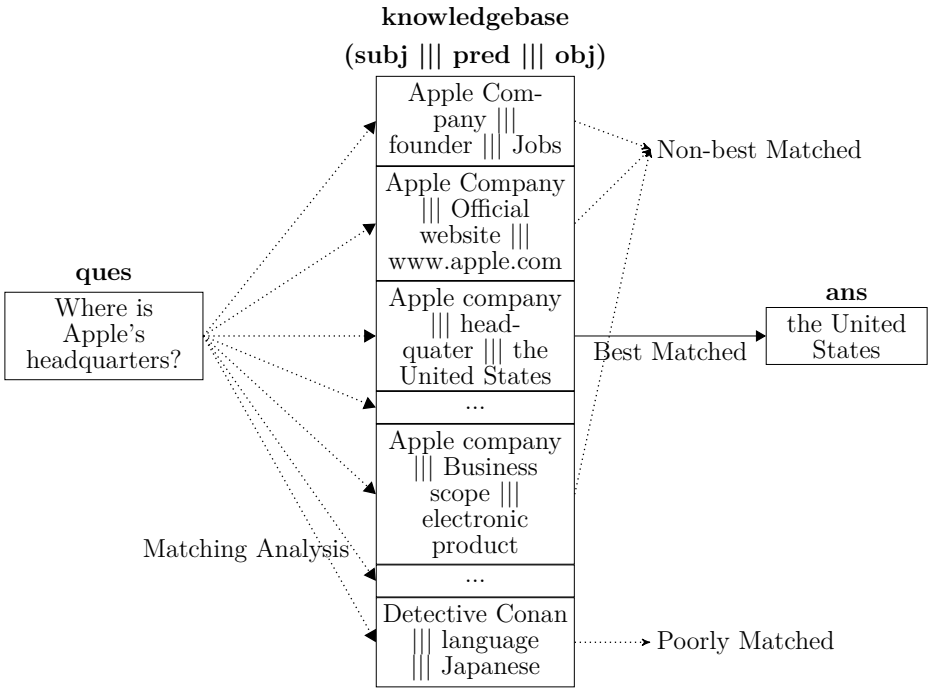

Figure 1.2: knowledge based question answering system

DBQA is more widely used, but KBQA is a research hotspot in recent years. The research content mainly focuses on knowledge representation, automatic construction of knowledge graph or knowledge bases, question information extraction, answer rankings, etc.

From the perspective of the corpus, the QA system can be divided into two categories: one is Open Domain QA System[33], and the other is Domain Specific QA System(also called by Restricted Domain QA System[34]). Open domain QA system can provide users common-sense answers from the unlimited field; Domain-Specific QA system is often used in a certain field with deep knowledge area. The open domain QA system is aimed at a wider user base, but it often requires a large knowledge base and technical support of the knowledgebase, and must deal with user problems in different language styles of spoken language. Domain Specific QA System aims at deep knowledge solutions for users in the

specific domain. Therefore, both QA systems have their own advantages and are widely used.

English QA technology started earlier and the corpus resources were richer. In contrast, Chinese QA technology started late, and due to Chinese language characteristics, open resources and research results are relatively few. The Chinese question answering system still has a lot of research worth studying. First of all, both English and Chinese face the problem of uncertainty in natural language. The language uncertainty under the QA system is mainly reflected in two aspects: one is the uncertainty of semantic expression, for example, the difference in the expression method of the question semantics, the uncertainty caused by the difference in the language style of the question; The other is the uncertainty of knowledge expression, for example, the different names of the same thing, the different descriptions of the same concept. Secondly, the Chinese language is different from English. For example, there are no obvious grammatical features such as spaces, which makes the English processing method not directly applicable to Chinese. Therefore, the uncertainty of the Chinese QA system should be solved in combination with the characteristics of the Chinese language.

1.2 Related Works

1.2.1 QA system

The difference between the question answering system and the traditional information retrieval is that, in the traditional information retrieval system user must input keywords, while in QA system the user inputs natural language question. Therefore, the QA system must first extract the search keywords or topic information from the sentences, and then get the most relevant answers. The QA process based on document retrieval can be transformed into the similarity matching based on the topic involved in the question. The basic method of topic similarity matching is the cosine similarity after the sentence is represented by word vector or sentence vector expression, such like VSM [35], LSI [36], LDA [37], Word2Vec[38], Doc2Vec[39] and other models.

The main principle of KBQA system is to match the question with the highest relevant knowledge resources, and then construct and return the answer. The knowledge

tuple usually uses a general description method called Resource Description Framework (RDF)[40] to describes the Resource-Attribute-Value (also called Resource-Property-Value or Subject-Predicate-Object) of knowledge. RDF descriptions are associated or stored in a sequence structure or graph structure, which constitutes the knowledge base [41] or the knowledge graph [42]. There are a large number of researches on QA systems based on English knowledge base or knowledge graph. The reference [43] proposes a method for extracting information based on the Freebase-based QA system, linking the key information in the grammar tree of the English question sentence with the key information on the knowledge graph; the reference [44] uses the convolutional neural network classifies the questions into three types (answer path, answer context, answer type) and then matches the links with Freebase.

There are also QA system frameworks that use web technologies or other sources to build knowledge links. The reference [45] introduces a method for mining knowledge from the web crawling information and used in the question answering system; the reference [32] proposed a knowledge QA method for expanding the knowledge base content and answer range through the network text content. The Chinese question answering system also has certain research results in the web-based question and answer [46, 47]. Most of the methods are to adapt the English grammar tree analysis method to Chinese.

Knowledge predicates and knowledge subjects can be seen as part of the topical features of short texts. In terms of short text topic feature extraction, The reference [48] introduces a topic classification method for Chinese short texts, which uses the high-frequency feature extension technology to optimize the LDA topic model; the reference [49] introduces a hot topic detection method for Chinese short texts; the reference [50] introduces a Chinese short text topic information monitoring method based on Ngram feature extraction and use Hownet to expand keywords to enhance short text feature performance.

1.2.2 Rough Set

Rough set theory is a kind of granular computing theory which is very suitable for dealing with uncertainty information. It was first proposed by Pawlak[51] and has been extended to many models such like fuzzy rough set [52], neighborhood rough set [53], variable precision rough set [54] and other models. The key concept of the rough set model is on the

upper approximation, the lower approximation and the boundary domain under different equivalence relations, and different degrees of granulation of knowledge by different equivalence divisions. Since different divisions offer different level of information granularity, and different information granularity leads to different level of uncertainty, rough set theory is an effective theory to deal with uncertain information such like language uncertainty and knowledge uncertainty.

The reference [55] proposes a similarity processing method for short texts based on rough sets, and uses rules acquisition methods to mine synonyms and polysemous words from texts; the reference [56] gives a text clustering model based on rough set. In the field of knowledge discovery, the reference [57] proposes a mining method for describing the logical concept of social networks based on rough set theory; the reference [58] proposes a customer feature discovery model based on rough set theory, which helps Three-party payment platform to tap potential customers; reference [59] uses rough set theory to extract knowledge from engine piston performance database and use it for smart motorcycle system design; reference[60] uses rough set theory to mine opinion knowledge and use it for public opinion prediction analysis.

Attribute reduction is a research hotspot of rough set theory. There is a large amount of related work on attribute reduction[61, 62]. One of the more classical reduction theories is Skowron's Discernibility Matrix Theory (DM Theory, also called separate matrix, discrimination matrix, distinguishing matrix, etc.)[63, 64]. The DM theory is usually used in the design of reduction algorithms, including both attribute reduction and value reduction:

In terms of attribute reduction, the document [65] defines a new DM and new discernibility function for the concept lattice, and proposes an attribute reduction algorithm based on the definition; the reference [66] proposes an incremental update based attribute reduction algorithm; reference [67, 68] proposed two different heuristic attribute reduction algorithms based on discernibility matrix; reference [69] proposed an attribute reduction algorithm based on a kind of binary discernibility matrix; the reference [70] proposes an attribute reduction algorithm based on the fuzzy discernibility matrix.

In terms of value reduction, the reference [71] introduces a value reduction algorithm based on the improved rule discrimination matrix; reference [72] proposes a heuristic value reduction algorithm based on the discernibility matrix.

1.2.3 Sentiment Analysis

Sentiment Analysis(SA) is one of the most important research topics in Information Processing, and within SA, the research on sentiment measurement and classification has been a very popular topic[73]. The main task in sentiment measurement and classification is to classify the text paragraph into different sentiment classes. Such technology has been applied in many fields, for example, opinion trend tracking of Twitter topics[74], customer reviews mining in marketing[75], and affective interaction between human and a dialogue robot[76, 77]. Most of the research is conducted at one of the three levels: document-level, sentence-level, or attribute-level[75], and at each level it involves different technologies at language representation methods and machine learning methods[78].

For some given applications such as Social Robots[76] and Twitter mining[74], the text length is usually very short and the sentiment analysis is at sentence-level in most time.

The first step of sentiment analysis is to change the input text into basic information representation such as tokens[79], POS(Part of Speech) tags[80], and parsing dependencies[81]. For analysis at sentence-level, POS tags and parsing dependencies are very useful since they contain much information about sentiment word positions and modifying targets[82]. Many language models focus on this kind of features using various data formats, for example, Knowledgebase[83][84](such as HowNet[85] and WordNet[86]), and corpus[87][88]. The statistical information[89][90] or other mathematical information[91] of features is also widely used in sentence preprocessing. For oral language text or network parlance processing other features such as using environment and text source also need to be considered, for example, Twitter hashtags and smileys[92] and user behavior[93].

Another aspect that may influence the sentiment tendency is the topic and opinion of the sentence. For example during a dialogue, the topic of celebrating a festival is mostly in positive sentiment class while the topic of earthquake is mostly in negative. Many language models for topic representation of short text have been proposed in recent years such as Biterm Topic Model (BTM)[94], and some technologies for long text topic mining are also developed into technologies for short text topic mining[95]. Some methods for predicting user's opinions can also be used in short text interaction[74][96].

There is also research work which focuses on sentiment processing of negative text. A

lot of methods have been proposed to detect the negative emotion and sentiment among large scale of data, such as news[97][98] and posts on Facebook [99][100]. Some methods for speech negative emotion have also discussed the text features of negation[101][102].

Since sentiment information usually depends on both semantic information and affective words, for a synthetic requirement a language model called SCLM[103] which can represent both semantic and sentiment information is first proposed for Social Robots. SCLM has some similarities with NaturalLI[104] at the idea of linguistic computing, and the matrix representation methods of the proposed language model resemble Recursive Matrix-vector Spaces[105] and other Neural Probabilistic Language Models[106][107]. The difference is that in SCLM the elements of matrices and vectors are based on sentence or paragraph level with speaker source and perspective coordinate information, while the others are based on word level and usually the perspective information is ignored at the model representation.

Supervised Machine Learning is a kind of Machine Learning where instances are given with known labels or the corresponding correct outputs[108]. Of all the Supervised Learning Methods, Supervised Neural Networks model[109][110][111] has become popular in recent years because of the development of Deep Learning(DL)[112][113][114], and Back-Propagation Neural Network(BPNN) is one of the most common supervised training algorithm[115, 116]. Deep Learning can be treated as an improved and much more complex version of NN, and it often deals with more complex information coding and encoding. In DL area, standard Recursive Neural Network(RNN)[117] model is the simplest NN based model, and based on RNN, Matrix-Vector RNN (MV-RNN)[105] and Recursive Neural Tensor Network(RNTN)[118] are developed. These three models have achieved good performance in language sentiment measurement. The other recent research on NN and DL has also got very significant achievements in Artificial Intelligent and Pattern Recognition[119], and they still have a lot of research space.

The advantages of the neural network method over traditional classifiers are its non-parametric nature, arbitrary decision boundary capabilities, easy adaptation to different types of data and input structures, fuzzy output values that can enhance classification, and good generalization for use with multiple images[120]. Considering the time and memory cost of training deep networks by Deep Learning method, in this thesis only the traditional

BPNN methods will be used to train our system for the first step of our research.

Chapter 2

Knowledgebase Construction for Short Text Processing

2.1 Introduction

The existing QA system is mainly divided into two types from the technical point of view: one is Document-based Question Answering system(DBQA system), that is, the returned answer or answer set which most matches the input question is based on a given set of candidate answers; the other is Knowledge-based Question Answering system(KBQA system), that is, without giving candidate sentences, the QA system will first get the answer information based on the QA knowledgebase, and use the answer information to construct the natural language answer and then return it as output. Both DBQA and KQBA are facing the difficulty of uncertainty expression processing for short text.

Uncertainty expressions of QA short text are mainly divided into two categories: the uncertainty of knowledge expression and the uncertainty of semantic expression. The two uncertainties are introduced separately below:

1. The uncertainty of QA knowledge content expression, that is, the content of questions and the corresponding answer content can be expressed in a variety of ways. For example, given the questions: 'What is the nationality of Leonardo?', 'Which country does Leonardo come from?', 'In which country Leonardo was born?', 'Is Leonardo a Chinese or an American?', the answer to these questions can be like: 'Leonardo

was born in America.', 'Leonardo is American.', 'Leonardo's birthplace is America.', and so on. Then the QA knowledge content can be expressed in many ways such as: 'Leonardo, nationality', 'Leonardo, birthplace', 'Leonardo, native land'. The uncertainty expression of QA knowledge content causes the difficulty when the system calculates the matching degree of question and candidate answers or knowledge items of the knowledgebase.

2. The uncertainty of QA semantic construction expression, that is, the question and answer sentences in the QA system can be expressed in a variety of ways in the process of constructing a complete natural language. For example, in some context, both the word 'Apple' and the phrase 'Apple Inc.' can refer to the Apple company. However, in the short text application environment, if the context is inadequate and the word 'Apple' is at the first place of a sentence, it will be difficult for the system to understand the true semantic object, and then will influence the knowledge or candidate matching accuracy of the system.

In this chapter, we mainly discuss about the uncertainty of knowledge content expression and propose a new knowledge construction method for short text processing. The knowledge representation method is based on the attribute reduction method and the upper approximation concept of the rough set theory. The proposed knowledgebase construction method can be used to discover the knowledge content from the labeled question and answer corpus and construct a rough knowledgebase which can deal with the uncertainty of QA short text. Then combined with the traditional sentence similarity model, the rough knowledge can be used to calculate the matching degree between the question and the candidate sentence in DBQA. The experimental results show that the new method is better than the traditional method on the two evaluation indicator of MAP and MRR.

2.2 Basic Concepts

Given the knowledge base $K = \{U, S\}$, where U is the domain and S is the equivalence cluster on U, then given $\forall X \subseteq U$ and $R \in IND(K)$, there are [121]:

$$\underline{R} = \{x | (\forall x \in U) \wedge ([x]_R \subseteq X)\} \tag{2.1}$$

$$\overline{R} = \{x|(\forall x \in U) \wedge ([x]_R \cap X \neq \emptyset)\} \tag{2.2}$$

$$Pos_R(X) = \underline{R} \tag{2.3}$$

$$Bn_R(X) = \overline{R} - \underline{R} \tag{2.4}$$

$$Neg_R(X) = U - \overline{R} \tag{2.5}$$

Wherein, the lower approximation $\underline{R}$ means the set of elements that certainly belong to X according to the equivalence relation R, and the upper approximation $\overline{R}$ means the set of elements that possibly belong to X, the boundary field $bn_R(X)$ indicates that it is temporarily impossible to determine whether it belongs to the X, and the negative field $neg_R(X)$ denotes a set of elements that are certainly not part of X.

2.3 Knowledge Discovery based on Rough Set

In a DBQA, the sentences in the corpus to be matched can be divided into three types:

1. Sentences that is almost irrelevant to the topic of the question, we call them the poorly matched sentences here,
2. Roughly matched the question, but still cannot be output as the best answer, we call them non-best matched sentences,
3. With the highest matching degree with the question, and can be used as the best output answers, we call it best matched sentences.

Most traditional methods can separate the poorly matched sentences from the other two sentences: the best matched and non-best matched sentences. However, due to the large amount of similar topic information, the traditional method will have a poorer discriminating ability to classify the best and non-best matched ones. Therefore, the method in this chapter eliminates the poor matched ones, and uses the remaining two sentences

as the candidate answer set, focusing on the discrimination between the non-best matched and the best matched.

Then the candidate answer set is correspondingly divided into two parts, one is the best matched answer set, which is called the positive sentence set denoted as Set_p, and the other is relative to the non-best matched answer set, called the negative matching sentence set, denoted as Set_n. The input question or questions is in a *ques* set. Each sentence in *ques* and Set_p , Set_n is treated as a collection of words. For each word, it can be divided into 7 categories according to the position in three kinds of sets. Marked by [*ques*, Set_p ,Set_n], the occurrence of words in sentences of question set, positive matched set, and negative matched set is shown in the table 2.1.

Table 2.1: Word tag and its meaning

No.	word tag	meaning
1	[0,0,1]	exists only in Set_n
2	[0,1,0]	exists only in Set_p
3	[0,1,1]	exists in both Set_p and Set_n
4	[1,0,0]	exists only in *ques*
5	[1,0,1]	exists in both *ques* and Set_n
6	[1,1,0]	exists in both Set_p and *ques*
7	[1,1,1]	exists in Set_p , Set_n and *ques* at the same time

When a question and a number of candidate sentences are given, the candidate answer selected into the positive match set satisfies two conditions:

1. The topic similarity of candidates and questions reaches the maximum value at the finest topic granularity. For example, the candidate answer ans_1 'Apple's boss is Jobs.' and the candidate answer ans_2 'Apple's headquarter is in America.', in the first level of topic granularity, the two candidates are both about 'Apple company', but at finer granularity, the topic of the former is 'Apple Company,boss', and the latter's topic is 'Apple Company, headquarter'. So if the question is 'Who is the owner of Apple', because the fine-grained topic of the question is 'Apple Company, boss', so the candidate ans_1 has a higher degree of matching than the other.

2. The candidate sentence contains information missing from the question under the same question topic granularity. For example, the candidate ans_1 in the above example can be the answer to the question, not only because it contains 'Apple' and 'Boss' at the same topic granularity as the question, but also it contains 'Jobs' as the answer information missing from this question.

The seven types of words in table 2.1 reflect the topic information and answer information at different granularities of the question and answer sentences. For example, the word 'Apple' exists in Set_p , Set_n and $ques$ at the same time and is marked as $[1, 1, 1]$, that means the QA topic is about 'Apple Company', while the word 'Who' marked $[1, 0, 0]$ means this word may only appear in question, and the word 'boss' marked $[1, 1, 0]$ may appears only in questions and positive sets and may be treated as the finest topic granularity.

The above process is a training process that uses different sets of sentences to determine the mark of a word. We can think of words as division rules. Differently labeled words are the degree to which a word can be divided into questions, positive matched and negative matched. The training process is the process of obtaining the division rules by training the text, and the retrieval (The test) process is a process of dividing candidate sentences into positive and negative matching sentence sets by using dividing rules and questions. According to the rough set theory, when given a question, a positive and a negative matched sentence set, the lower approximation words of the QA knowledge formed by the topic of the question and the corresponding answer information are more likely to be marked as [1,1, 0], [1,0,0], [0,1,0] in the words set, and the words labeled [0,0,1] is the negative domain of question and answer knowledge.

After getting the categories through the training questions and the positive and negative matching set words, we only remove the stop words, commonly used punctuation, and the negative words of the topic category marked as [0,0,1], using the remaining words set and the corresponding mark to represent the upper approximation of a QA knowledge content.

There are two kinds of special training situations: one is the given candidate answer sentences are all positive matching sentences, then the negative matching sentence set is

an empty set. At this condition, the rough QA knowledge obtained by training does not include [0,1,1], [1,0,1] and [1,1,1] words; the other is that, candidates are all negative match sentences, but since our training goal is to find out the correct topic and answer knowledge of QA sentences, these training samples need to be eliminated before training.

2.4 Text Retrieval by Rough QA Knowledge

After training to obtain a series of rough set question and answer knowledge, when the QA system obtains new questions and candidate answers, the matching degree between the questions and the answer can be calculated by the formula 2.6:

$$QAM = \alpha \cdot SSim + \beta \cdot KMatch \tag{2.6}$$

$SSim$ is the traditional sentence similarity of a question and an answer candidate, calculated by the traditional vector model such as LDA, LSI; $KMatch$ is the rough knowledge similarity, α and β is the weight.

The process of calculating $KMatch$ is as shown in the algorithm 2.1 and the algorithm 2.2:

Algorithm 2.1 $KMatch$ Computing(1)

Input: question, answer candidate sentences

Output: Maximum similarity of hypothetical knowledge for all candidate answers

1: Select a sentence from the candidate answer sentences, first assume that it is a positive matched sentence, and the others are negative. Mark all the words with the assumed knowledge through the training process, and then remove [0, 0, 1] words. Then we get the hypothetical QA knowledge item by the hypothetical positive sentence;

2: Calculate similarity between the hypothetical QA knowledge and the real rough QA knowledgebase items, using the algorithm of 2.2.

3: Iterate through all the candidate sentences to get the maximum similarity of the hypothetical knowledge of all candidate sentences.

Algorithm 2.2 $KMatch$ computing(2)

Input: the hypothetical QA knowledge, the real rough QA knowledge-base items

Output: The maximum similarity of the candidate sentence relative to the hypothetical knowledge

1: Compare all the lexicons of the hypothetical category knowledge and one item of the rough set knowledgebase. If the words and the marks are the same, the number of corresponding tokens is increased by 1. By the order of [[0,1,0], [0,1,1], [1,0,0], [1,0,1], [1,1,0], [1,1,1]], we can get a count vector with a dimension of 6 A.

2: Compare the total count of A and the threshold value C, if less than C, return $KMatch = 0$ and go step 5, otherwise to step step 3.

3: Calculate the total count of position $ques$ and Set_p of [$ques$, Set_p , Set_n], if anyone of them is 0 then return 0 and to step 5, otherwise go step 4.

4: Normalize the count vector A and a given hypothetical average knowledge vector K, then use the cosine similarity formula to calculate their similarity and go step 5.

5: Repeat step 1~4 and return the maximum similarity of the candidate sentence relative to K.。

2.5 Experiment

The experiment uses the open data set and evaluation tool of the DBQA sub-task of the open domain QA system in the NLPCC-ICCPOL2016[27] international conference. The dataset includes two parts: the training set and the test set, the training set contains 8772 questions and 181882 candidate answers; the test set has a total of 5,997 questions, and 122,531 candidates. All questions and answers do not involve contextual semantics.

The experiment uses the same MRR (Mean Reciprocal Rank) and MAP (Mean Average Precision) evaluation indicators as the evaluation competition. Among them, the formula

for calculating MRR is:

$$MRR = \frac{1}{Q}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{2.7}$$

$|Q|$ is the total number of questions, and $rank_i$ is the rank of the first correct answer in the candidate answer for the i question. If there is no correct answer, let $\frac{1}{rank_i}$ be 0.

MAP formula is:

$$MAP = \frac{1}{Q}\sum_{i=1}^{|Q|}AveP(C_i, A_i) \tag{2.8}$$

$$AveP(C_i, A_i) = \begin{cases} \frac{\sum_{k=1}^{n}(P(k)\cdot rel(k))}{min(m,n)}, & min(m,n) \neq 0 \\ 0, & min(m,n) = 0 \end{cases} \tag{2.9}$$

m is the correct number of positive matches, n is the number of positive matches given by the system. If $min(m, n)$ is 0, then $AveP(C_i, A_i)$ is 0. If the system gives a candidate sentence with a ranking of k that is the correct positive match, then $rel(k)$ is 1, else 0. $P(k)$ is the proportion of the correct positive matching sentence in the first k candidate sentences given by the system.

In the experiment, the cosine similarity method after vectorization by the traditional LSI model is used as the baseline1 in the comparison experiment (LSICosine), and the cosine similarity method by LDA model is used as the baseline2 (LDACosine), the cosine similarity by Doc2Vec model is used as baseline3 (D2VCosine). All baseline experiments are implemented using the Gensim toolkit [122]. Since the LDA model performs best on the corpus, the score by LDA model is used in the proposed method to calculate $SSim$ value of QAM.

When K = [2,1,2,1,4,1] and $C = 2$, the results are:

Table 2.2: results by K=[2,1,2,1,4,1],$C = 2$

	MAP	MRR
LSICosine	0.5372	0.5376
LDACosine	0.6386	0.6392
D2VCosine	0.3290	0.3300
RKMethod	**0.6449**	**0.6457**

The experiment results show that the method combining rough set knowledge is improved compared with the other three baseline methods in the two evaluation indicators of MAP and MRR, and improves the effectiveness of the proposed method.

If fix K to [1,1,1,1,1,1] at first, then increase the weight of the elements at each position in steps of 0.5 (for example:[1,1,1,1,1,1],[1.5,1,1,1,1,1],...,[6,1,1,1,1,1]),The effects of increasing the weight on each position on the final experimental results are tested in turn, and the changes in MAP and MRR are shown as 2.1 ~2.6:

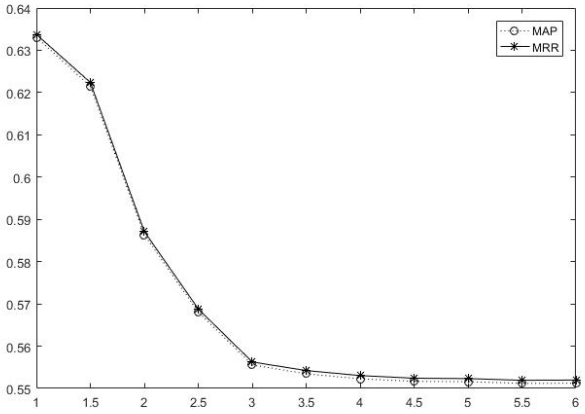

Figure 2.1: [0,1,0]

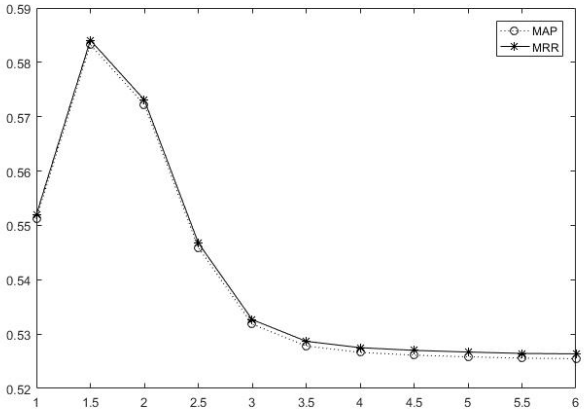

Figure 2.2: [0,1,1]

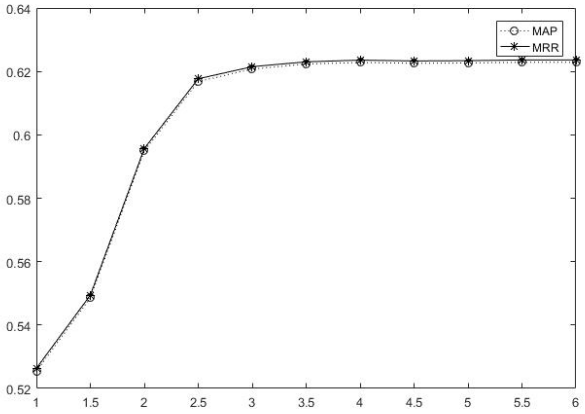

Figure 2.3: [1,0,0]

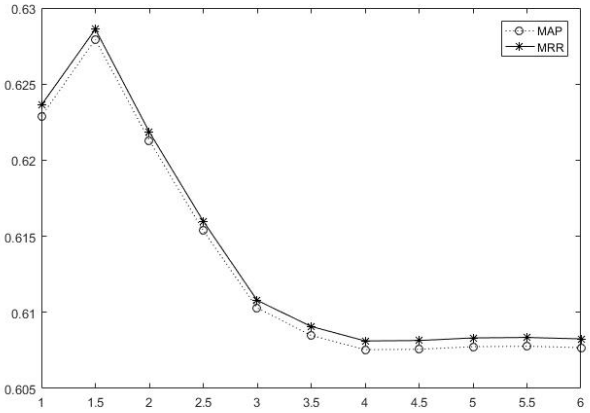

Figure 2.4: [1,0,1]

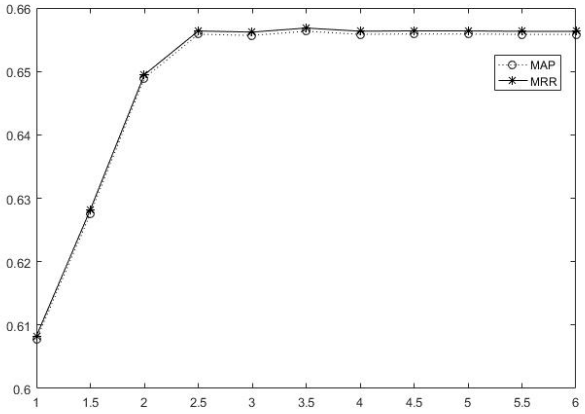

Figure 2.5: [1,1,0]

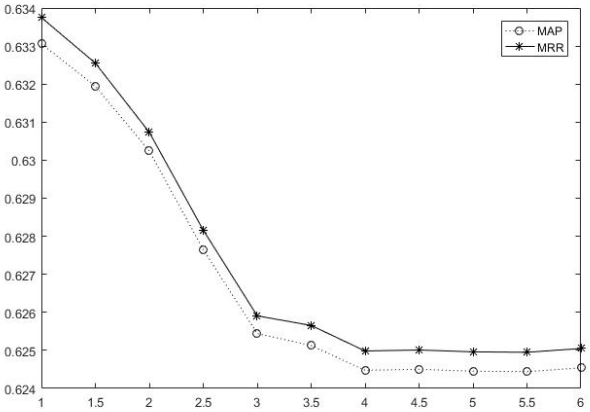

Figure 2.6: [1,1,1]

After increasing the relative weights of [1,0,0] words and [1,1,0] words, the MAP and MRR values also increase especially in the initial stage. That means, the more common topic words in the interrogative words, the question and positive matched sentence, the easier to get the required QA knowledge.[0,1,0] words and [1,1,1] words will always decrease the MAP and MRR values as the weight increases, that means: if a word only appears in positive matched set, it may be missing the topic or knowledge information required by the question, because it is just the uncommon semantic expression in a positive matching sentence; or, if a word appears in three kinds of the sentences at the same time, then it does not have the ability to distinguish both the question and the positive matched from the negative matched. Words with other marks will have a peak at the beginning, but

overall the effect on the result is a decline. It can be seen that the experimental results are basically in line with reality.

If K = [2,1,2,1,4,1] is fixed, the filter threshold C is gradually increased from 1 in steps of 1 , the values of MAP, MRR, and timecost for once traverse are shown in 2.7.

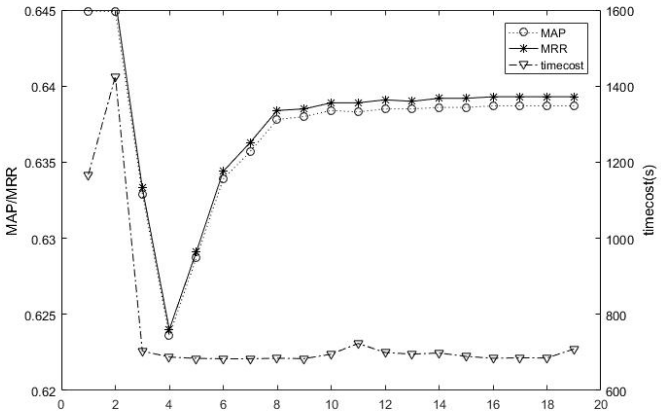

Figure 2.7: Experiment results when increase C

It can be seen that when the filtering thresholds are 1 and 2, MAP and MRR values are the highest, but the relative timecost is also relatively long. The time required to traverse a test data set for once is 1400~1600s. However, when the threshold exceeds 3, the timecost is greatly reduced, only about 700s, which is about half of the time taken when the threshold is 1 and 2. As the threshold increases, the timecost does not change significantly. When the threshold exceeds 8, the MAP and MRR values do not fluctuate greatly. The MAP value remains within the range of [0.6378, 0.6387], and the MRR value remains at [0.6384, 0.6393], the highest value in the two intervals is only 0.0001 higher than the highest MAP and MRR values obtained from the three baseline experiments, but timecost is higher than the baseline because of the knowledge matching process. Therefore, after considering both time and effect, it is most appropriate to select 1 or 2 for the filtering threshold.

2.6 Conclusion

The advantage of the proposed method is that it can mine out potential knowledge representation information from multiple sets of positive and negative matched sentences in

training corpus. But there is also improvement space in the following areas:

1. In the experiment, the parameters such as K,C ,α and β is set by experience. How to set parameters more objectively according to the actual application is a problem that needs to be solved. and also , in $KMatch$ calculating process, the words counting method is also one of the follow-up studies.

2. The data set in this experiment is in the form of a single question, a positive matched set of one or more sentences, and a negative matched set of one or more sentences. Therefore, the rough QA knowledge can only find out the potential answer expression from the answer candidates. In the future work, we can try to expand the data set by a synonym sentence set, positive matched sentence set and negative matched sentence set. By adding synonym sentence questions it will be easy to find out more potential question expressions.

Chapter 3

Rules Acquisition for Short Text Processing

3.1 Introduction

In this chapter we will introduce a novel method for Short Text and Information Retrieval based short text Question Answering. Based on the Rough Set Theory and Discernibility Matrix based Rules Acquisition method, the matching patterns of the training QA pairs can be represented as rules by the reduced attribute words, and the words can also be represented by the QA patterns. Then the attribute words in the test QA pairs can be used to calculate the matching scores. The experimental results show that the proposed representation method of QA patterns has good flexibility to deal with the uncertainty caused by the short text word segmentation, and the proposed method has good performance at both MAP and MRR on the test data.

3.2 Basic concepts

In Rough Set Theory, a decision table [123] is defined as *Formula* (3.1).

$$DecisionTable = \{U, A = C \cup D, V, f\} \tag{3.1}$$

In a decision table, U is a finite nonempty set of objects, and A is a finite nonempty set of attributes of the objects. A is divided into two subsets, where one is the set of

condition attributes and the other is the set of decision attributes. V is a nonempty set of values of all the attributes, and $f : U \times A \to V$ is the function that maps an object of U by a attribute of A to a value of V. If there are two objects having the same values of all the condition attributes but their decision attribute values are different, the decision table is inconsistent; otherwise it is consistent.

Based on a decision table, we can get its $POS_c(D)$ by *Formula* (3.2) and *Formula* (3.3). $POS_C(D)$ is called a positive region of the partition U/D with respective to C, and is a set of all elements of U that can be uniquely classified to blocks of the partition U/D, by means of C. C_*X is called the $C - lower$ region of X, and $C(x)$ is the equivalence class containing an element x.

$$POS_C(D) = \bigcup_{X \in U/D} C_*X \tag{3.2}$$

$$C_*X = \{x \in U | C(x) \subseteq X\} \tag{3.3}$$

Sometimes not all the condition attribute are necessary. If a condition attribute $c \in C$ satisfies *Formula* (3.4), c is not necessary and can be reduced.

$$POS_{\{C-c\}}(D) = POS_C(D) \tag{3.4}$$

A lot of Rough Set Theory based methods have been proposed for attribute reduction [62]. Our proposed method for QA system is developed from the discernibility matrix theory [63]. The classical discernibility matrix is a $|U| \times |U|$ matrix, and its element $M(x, y)$ defined as *Formula* (3.5). Based on the discernibility matrix, we can get the discernibility function by *Formula* (3.6).

$$M(x, y) = \{a | a \in A, f(x, a) \neq f(y, a)\} \tag{3.5}$$

$$df(M) = \wedge\{\vee(M(x, y)) | M(x, y) \neq \emptyset\} \tag{3.6}$$

3.3 Rules Acquisition and Attribute Vectorization

In this section, we will introduce the training processing of our method, including rules acquisition of short text QA sentences and vector representations of the attribute words. The attribute word representations are based on the rules, and the representations will be used for matching QA patterns in the testing processing.

3.3.1 Rules Acquisition of short text QA Sentences

Given one question and m labeled candidate items (all the sentences have been segmented into words, the label means whether the item can be used as a answer of the question or not), we first construct a dictionary of all the words of the question and the items. For convenience, we name the item which can match the question as Positive Sentence (PS), and the other Negative Sentence (NS). We name the set of all the PS as Positive Sentence Set (PSS) and the other Negative Sentence Set (NSS). After we get the dictionary, we first remove the words which appear only in the NSS, and also remove some short text stopwords. This pre-filtering step will help reduce the dimension and accelerate the following attribute reduction and rules acquisition, and can also make each of the final rule attribute words appear at least once in a PS or the question.

Using the dictionary of n words, we can construct a small Question Answering Matching System (QAMS) for the question and its candidate items, like *Table* 3.1. We define this small decision system as $QAMS = \{U = I \cup Q, A = W \cup D, V = 1, 0, f\}$. $I = \{I_1, I_2, ..., I_m\}$ is the candidate items set, and Q is a set with only one question in it. $W = \{w_1, w_2, ..., w_n\}$ is the word attribute set (the dictionary), and D is the decision attribute set with only the matching label attribute in it. The function $f(u, a)$ is defined as *Formula* (3.7).

$$f(u \in U, a \in A) = \begin{cases} 1, & \text{if } a \in D \text{ and } u \in PSS \cup Q; \\ & \text{or if } a \in W \text{ and } a \in u \\ 0, & \text{the other} \end{cases} \tag{3.7}$$

The function $f : U \times A \to V$ means that if an attribute word appears in an item or the question, and the attribute value equals 1, or if the item is a PS or the question, its

Table 3.1: Question Answering Matching System (QAMS)

Item/question	w_1	w_2	...	w_n	Decision Label
I_1	v_{11}	v_{12}	...	v_{1n}	v_{1l}
I_2	v_{21}	v_{22}	...	v_{2n}	v_{2l}
...	...	...	...	...	...
I_m	v_{m1}	v_{m2}	...	v_{mn}	v_{ml}
question	v_{q1}	v_{q2}	...	v_{qn}	v_{ql}

decision attribute value is 1. Then we need to mining the rules in the $QAMS$. Since for QA system, we only need to concern about the rules for question and its PSS. Then the discernibility matrix of the $QAMS$ is a $x \times y$ matrix, $x = |PSS| + |Q|$, $y = |NSS|$. The values of the QA Discernibility Matrix (QADM) is defined as $Fomular$ (3.8) and $Fomular$ (3.9).

$$Dset(u_p, u_n) = \{a | a \in W, u_n \in NSS, u_p \in PSS \cup Q, f(u_p, a) \neq f(u_n, a)\} \tag{3.8}$$

$$QADM(u_p, u_n) = \begin{cases} Dset(u_p, u_n), & \text{if } |Dset(u_p, u_n)| > 0 \\ \{a | a \in u_p\}, & \text{the other} \end{cases} \tag{3.9}$$

The discernibility function of the $QADM$ is defined as $Fomular$ (3.10).

$$df(QADM) = \wedge \{\vee(QADM(u_p, u_n)) | u_n \in NSS, u_p \in PSS \cup Q, QADM(u_p, u_n) \neq \emptyset\} \tag{3.10}$$

In the function expression of $QADM$, $\vee(QADM(u_p, u_n))$ is the disjunction of all attributes in $QADM(u_p, u_n)$ and $\wedge\{\vee(QADM(u_p, u_n))\}$ is the conjunction of all $\vee(QADM(u_p, u_n))$. When u_p and u_n is inconsistent, that is to say, all of their attribute words are the same, we will set the value of $QADM$ by the attributes of u_p. The original corpus of QA system is consistent theoretically. However, there are two reasons for this definition: one is that it can avoid the error case of the mislabeled items in the corpus, and the other is that

Table 3.2: An Example of a QAMS

Item/question	w_1	w_2	w_3	w_4	Decision Label
I_1	0	0	1	1	1
I_2	1	0	1	0	1
I_3	0	1	1	1	0
I_4	0	1	0	1	0
question	1	1	1	1	1

Table 3.3: The QADM of the QAMS in Table 3.2

	I_3	I_4
I_1	$\{w_2\}$	$\{w_2, w_3\}$
I_2	$\{w_1, w_2, w_4\}$	$\{w_1, w_2, w_3, w_4\}$
question	$\{w_1\}$	$\{w_1, w_3\}$

after the pre-filtering step the consistent $QAMS$ may turn to inconsistent.

A $QAMS$ example is showed in $Table$ 3.2 and its $QADM$ is showed in $Table$ 3.3. The example $QAMS$ is with 4 attribute words and 4 candidate items. 2 of the 4 candidate items are PSs. The $QADM$ of it is a 3×2 matrix. Based on $Formula$ (3.10) we can get the discernibility function, showed in $Formula$ (3.11). The result of $Formula$ (3.11) means that a question and its PSs can be discerned from the NSs by the words w_1 and w_2.

$$\begin{aligned} df(M) &= (w_2) \wedge (w_2 \vee w_3) \\ &\quad \wedge (w_1 \vee w_2 \vee w_4) \wedge (w_1 \vee w_2 \vee w_3 \vee w_4) \\ &\quad \wedge (w_1) \wedge (w_1 \vee w_3) \\ &= (w_1) \wedge (w_2) \end{aligned} \tag{3.11}$$

If the result is like $(w_1 \vee w_3) \wedge (w_2)$, that means the discernibility rules can be w_1 and w_2, or can be w_3 and w_2.

3.3.2 Vector Representation of Attribute Word

Given a set of questions and their labeled candidate items, we can get all of their $QAMS$s, reduced attributed words and rules. Based on the reduced attribute words and the acquired rules, each of the attribute words can be represented as list of vectors. The vector unit v is defined as $Formula$ (3.12). $NO.(QADM)$ is the number label of the $QADM$, $Len(df_{QADM})$ is the sum count of all conjunction elements in the final result of the discernibility function, and $NO.(w_{df})$ is the number label of the conjuncted element of the final result in which the word appears. $T(w_{df})$ is the tag whether the word is appeared in the question or candidate items or both of them.

$$v = [NO.(QADM), \quad Len(df_{QADM}), \quad NO.(w_{df}), \quad T(w_{df})] \tag{3.12}$$

After we trained a set of questions and its labeled candidate items, all the attribute words can be represented like $Formula$ (3.13). In this Formula, θ is the appearance times of the attribute words in all the $QADM$ of the corpus.

$$WV = [v_1, v_2, ..., v_\theta] \tag{3.13}$$

For example, if the $QAMS$ is the second one of the whole training corpus and the word w_1 and w_2 does not appears in other $QAMS$s, based on $Formula$ (3.11) the word w_1 can be represented as $Formula$ (3.14) and the word w_1 can be represented as $Formula$ (3.15). The ellipsis is the cases of the word appearance vectors in other $QAMS$s.

$$WV_{w_1} = [\quad [2, 2, 1, \{\text{'Q','PSS'}\}] \quad , ...] \tag{3.14}$$

$$WV_{w_2} = [\quad [2, 2, 2, \{\text{'Q'}\}] \quad , ...] \tag{3.15}$$

The attribute words and the acquired rules can be treated as a kind of QA sentence patterns, and $NO.(QADM)$ can be treated as the QA pattern number. However, the model lacks the topic information of the QA. So when it comes to practical application, it must be used at the same time with some topic similarity model.

Table 3.4: An example of the middle dictionary of the patterns

$NO.(QADM)$	vlist	$Len(df_{QADM})$	vlistlength	C_{QADM}
36	{[36, 4,2, {'Q'}] , [36, 4,1, {'PSS'}]}	4	2	0.5
53	{ [53, 1,1, {'Q','PSS'}] }	1	1	1
...	...	...	...	...
182	{[182, 4,2, {'Q'}] }	4	1	0

3.4 Method of Matching QA Patterns

We can get a dictionary with all the attribute words represented by *Formula* (3.13). Then when a test question and an unlabeled candidate item are given, we can get two list of word vector elements from the attribute words appears in the two word sequence: $VL_q = [v_1, v_2, ...]$ and $VL_{I_i} = [v_1, v_2, ...]$. The next step is to count up the QA pattens and measure their completeness. But before that we must do some preliminary reduction.

At the reduction step, there are two kinds of processing choices. One is that we need to concern the word vector element tag $T(w_{df})$, that means, for example, if a word appears only in the question, and one of its vector tag means it appears only in the NSS in a QA pattern of the train corpus, we must remove it from VL_q. That means we treat strictly that in one QA pattern, the word role of it should not be exchanged. The other processing choice is that we just ignore the tags and we consider that sometimes the words among question and candidate items can be exchanged and will not change the semantic too much.

Then based on the $NO.(QADM)$ we count up the pattern and its vector elements (the same elements are counted only once). An example of the middle dictionary of the patterns is illustrated in *Table* 3.4. Here we define the completeness of a pattern (QADM) as *Formula* (3.16).

$$C_{QADM} = \begin{cases} 0, & \text{if } vlistlength = 1 \text{ and } Len(df_{QADM}) \neq 1 \\ \dfrac{vlistlength}{Len(df_{QADM})}, & \text{the other} \end{cases} \tag{3.16}$$

and the final completeness of the QA pairs is calculated by *Formula* (3.17).

$$C(q, I_i) = \sum_{\bigcup QADM|q,I_i} C_{QADM} \tag{3.17}$$

3.5 Experiment

The experiment is divided into two parts: one is on the sentence pattern similarity and the other is on the text retrieval. As there are two choice at the reduction step of the Matching method (with vector tags and without tags), we evaluate both in the experiment. The first experiment is comparing the proposed method with the word2vec pattern similarity method, and in the second experiment it is compared with cosine similarity of LDA and LSI model. In the second experiment, the text similarity matching part of our method is the same as LDA baseline.

Both the two experiments use the opensource corpus and toolkits of NLPCC-ICCPOL2016 Shared Task (Evaluation Competition) [27]. The corpus contains a train subset and test subset. The train set contains 8772 question texts, and the test set contains 5997 questions. Each of the question is given a list of candidate items and some of the items can be used as answers to the question. The train set contains 181882 items and the test set contains 122531 items. The baseline models of the experiments are constructed by Gensim Toolkit [124], and the word segmentation of all the short text text is completed by the NLPIR (also named as ICTCLAS) tool [125].

In our experiment, the evaluation metrics is the same with the competition:Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). MAP and MRR have been introduced in the chapter 2.

The experimental results are in *Table* 3.5 and *Table* 3.6. In *Table* 3.5, the withtags version of our method has best performance, but the withouttags version is not unsatisfactory. In *Table* 3.6, both the two version of our method have improve the performance of LDA baseline, and they all have better performance that LSI baseline model.

The MAP and MRR results of the withtags version of our method are higher than the withouttags version at both of the two experiments. It shows that at this QA corpus, most of the attribute words have fixed roles in QA patterns. So the final rule expres-

Table 3.5: Results of Sentence Patterns Similarity Experiment

	MAP	MRR
W2Vcosine	0.4075	0.4081
DM(withtags)	**0.4520**	**0.4525**
DM(withouttags)	0.2923	0.2924

Table 3.6: Results of QA Retrieval Experiment

	MAP	MRR
LDAcosine	0.6386	0.6392
LSIcosine	0.5372	0.5376
DM(withtags)	**0.6464**	**0.6469**
DM(withouttags)	**0.6436**	**0.6440**

sions acquired by the withtags version method can represent more information of the QA patterns.

3.6 Conclusion

In this chapter a novel method for short text and Information Retrieval based short text Question Answering is proposed. It has good flexibility to deal with the short text QA uncertainty by mining and representing QA pattern, and the proposed method has good performance at both MAP and MRR on the test data. The future work will focus on more QA experiments by other kinds of feature selection and attribute reduction method based on Rough Sets and on other short text QA corpus.

Chapter 4

Subject Analysis for Knowledge Triple

4.1 Introduction

In the KBQA system, when the system receives the question, it will separately analyze out the words containing the knowledge subject and the knowledge predicate, and then match the corresponding positions of the knowledge tuples in the knowledgebase or knowledge graph to select the knowledge tuple with the highest matching degree. The accurate extraction of knowledge subject is not only one of the important processes affecting the matching accuracy of the KBQA system, but also one of the important processes of knowledge subject positioning based on the knowledge map.

However, knowledge subjects cannot be directly replaced by traditional sentence subjects or named entities. The table 4.1 gives a Chinese short text example (by Stanford-CoreNLP, the language model used by the tool is the official Chinese processing model [16]). We can see that the subject of the traditional syntax given by the analysis tool is ”you”, the subject in the clause is ”institution”, the named entity is ”Portuguese”, while the knowledge tuple corresponding to the sentence in the knowledgebase is ” [subj: Portuguese teaching, predicate: management institute, obj: International Portuguese Language Institute], the corresponding knowledge subject is Portuguese teaching. This difficulty is very common especially in Chinese short text because of flexible characters and words expression of Chinese language.

Table 4.1: Three processing results of a sentence

Chinese sentence	你 (you) 知道 (know) 是 (is) 什么 (what) 机构 (institution) 在 (-ing) 管理 (manage) 葡萄牙语 (Portuguese) 教学 (teaching) 吗 (*interrogative auxiliary*)
English translation	Do you know what institution is managing Portuguese teaching
list of (word segments, POS, No.) series	(' 你', 'PN', 1), (' 知道', 'VV',2), (' 是', 'VC', 3), (' 什么', 'DT', 4), (' 机构', 'NN', 5), (' 在', 'P', 6), (' 管理', 'NN', 7), (' 葡萄牙语', 'NN',8), (' 教学', 'NN', 9), (' 吗', 'SP',10)
Dependencyparsing result	('ROOT', 0, 2), ('nsubj', 2, 1), ('ccomp', 2, 3), ('det', 5, 4), ('root', 3, 5), ('case', 9, 6), ('compound:nn', 9, 7), ('compound:nn', 9, 8), ('nmod:prep', 5, 9), ('dep', 9, 10)
NER tagging	(' 你', 'O'), (' 知道', 'O'), (' 是', 'O'), (' 什么', 'O'), (' 机构', 'O'), (' 在', 'O'), (' 管理', 'O'), (' 葡萄牙语', 'DEMONYM'), (' 教学', 'O'), (' 吗', 'O')

In this chapter we proposes a method for extracting knowledge subject from questions based on the attribute importance degree of rough set theory, and combines the existing sequence labeling method for named entity recognition and syntactic subject analysis to label the results for knowledge subject extraction, and then improve the overall knowledge subject analysis ability of the system.

4.2 Model Training

Given a training corpus, it includes a series of questions and the correct knowledge subject extraction results for each question. For each of these training questions *ques*, using NLP tools to analyze the word segmentation and its dependent grammar information; For each

word segment, label the knowledge subject extraction mark according to the content of the knowledge subject and get all dependent tuples. Tabel4.2 shows an example of this pre-process(by Stanford CoreNLP, the Chinese sentence means ‘How many pages are there in the ”University Computer Basics Tutorial”?’):

Table 4.2: example of preprocess

NO.1	segment	Label	*dplist*	L_{dplist}
1	《	BLeft	{ [’punct’, ’ 基础’, ’ 《’]}	1
2	大学	KSubj	{ [’compound:nn’, ’ 基础’, ’ 大学’]}	1
3	计算机	KSubj	{ [’compound:nn’, ’ 基础’, ’ 计算机’]}	1
4	基础	KSubj	{ [’punct’, ’ 基础’, ’ 《’], [’compound:nn’, ’ 基础’, ’ 大学’], [’compound:nn’, ’ 基础’, ’ 计算机’], [’nsubj’, ’ 有’, ’ 基础’], [’punct’, ’ 基础’, ’》 ’]}	5
5	》	BRight	{ [’punct’, ’ 基础’, ’》 ’]}	1
6	一共	Neg	{ [’advmod’, ’ 有’, ’ 一共’]}	1
7	有	Neg	{ [’ROOT’, 0, ’ 有’], [’nsubj’, ’ 有’, ’ 基础’], [’advmod’, ’ 有’, ’ 一共’], [’dep’, ’ 有’, ’ 多少’], [’punct’, ’ 有’, ’? ’]}	5
8	多少	Neg	{ [’dep’, ’ 有’, ’ 多少’], [’mark:clf’, ’ 多少’, ’ 页’]}	2
9	页	Neg	{ [’mark:clf’, ’ 多少’, ’ 页’]}	1
10	?	Neg	{ [’punct’, ’ 有’, ’? ’]}	1

dplist is a list of dependencies involved in each word,L_{dplist} is the length of *dplist*. ’BLeft’ labels the first left word of the knowledge subject start word, ’Bright’ labels the first right word of the knowledge subject end word, ’KSubj’ labels the knowledge subject part of the sentence, and ’Neg’ labels the rest words. In some special cases, the boundary markers ’BLeft’ and ’BRight’ may not correspond to the participles in the actual sentence. Therefore, when the sequence number of the first word of a sentence is 0, if the knowledge subject part starts from the beginning of the sentence, set the BLeft number to -1; if the

knowledge subject part ends from the end of the sentence, set the BRight number to the total word number in the sentence.

By the above processing on all the questions in the training corpus, we can get a preliminary rule dictionary. In the dictionary, the data of the dictionary value DV corresponding to each word is stored in the structure shown in table 4.3:

Table 4.3: structure of dictionary value DV

No.	$dplist$	count of samples	tag of word
1	dpl_1	C_{dpl_1}	t_1
2	dpl_2	C_{dpl_2}	t_2
...	...	...	...
n	dpl_n	C_{dpl_n}	t_n

The dependency relationship contained in each dplist in table 4.3 is used as the condition attribute of the rough set decision table, the word mark is used as the decision attribute, and the above table is converted into the decision table DT under the rough set theory. By rough set theory, if a decision table has rules which has the same conditions but different decisions, then the decision table is inconsistent; or the decision table is consistent[64]. Then we can divide the decision table into two parts: the consistent part DT_c and the inconsistent part DT_{ic}. The division process is as shown in the algorithm 4.1.

Algorithm 4.1 division processing of DV

Input: DV

Output: DT_c, DT_{ic}

initialize DT_c, DT_{ic}
for dpl_i in DV **do**
  if $\exists$ dp_j: dp_j==dp_i **and** $t_j \neq t_i$ **then**
    $DV = DV$ - dp_j - dp_i
    $DT_ic = DT_ic$+ dp_j + dp_i
  else
    $DV = DV$ - dp_i
    $DT_c = DT_c$+ dp_i
  end if
end for

According to the consistent table DT_c and the inconsistent table DT_{ic}, the processing continues to obtain two attribute sets, which are called consistent attribute set A_c and inconsistent attribute set A_ic. The two types of attribute sets reflect some extent the ability of attribute words to express decision-making concepts. The process is shown in the algorithm 4.2.

Algorithm 4.2 division processing of attribute set

Input: DT_c, DT_{ic}

Output: A_c, A_{ic}

initialization ofA_c, A_{ic} i
for $dpl_i \in DT_{ic}$ **do**
  for a_i in dpl_i **do**
    $A_{ic} = A_{ic} + a_i$
  end for
end for
for $dpl_i \in DT_c$ **do**
  for $a_i \in dpl_i$ **do**
    if $\exists\ dp_j \in DT_{ic} : a_i \in dp_j$ **then**
      $A_{ic} = A_{ic} + a_i$
    else
      $A_c = A_c + a_i$
    end if
  end for
end for

The importance of the attribute reflects on the one hand the tendency of misjudgment of the decision table after the attribute is removed, and on the other hand reflects the ability to make effective judgments on the items in the domain through this attribute. In the inconsistent decision table, the condition attribute of the inconsistent item is temporarily unable to correctly divide the decision of the project because the context still lacks a certain number of necessary judgment conditions. Therefore, on this basis, the definitions of the algorithm 4.3 and the formula 4.1 are given.

Algorithm 4.3 $ec(a_i, DT_c)$ computing

Input: DT_c, $a_i \in A_c$,

Output: $ec(a_i, DT_c)$

1: copy a DT_c to DT_{tmp}

2: initialize $ec(a_i, DT_c) = 0$

3: remove the a_i of $dpl \in DT_{tmp}$ and get DT'

4: **for** $dpl_i \in DT'$ **do**

5: **if** $L_{dpl_i} == 0$ **or** $(\exists dpl_j : dpl_j == dpl_i \textbf{and} t_j \neq t_i)$ **then**

6: $ec(a_i, DT_c) = ec(a_i, DT_c) + C_{dpl_i}$

7: **end if**

8: **end for**

$$Importance(a_i) = \begin{cases} \dfrac{ec(a_i, DT_c)}{|DT_c|}, & \text{if } a_i \in A_c \\ 0, & \text{the other} \end{cases} \tag{4.1}$$

In the formula 4.1, $ec(a_i, DT_c)$ reflects the proportion of the number of items in the decision table that have failed to make decisions in the overall consistent decision sub-item after removing the attribute a_i. If A_i does not appear in the consistent decision table, on the one hand, a_i lacks some necessary information to assist and will lead to wrong decision making, on the other hand, a_i does not have the strength to support decision making.

After obtaining the consistent attribute set A_c and the inconsistent attribute set A_{ic}, calculate the importance of each attribute using the algorithm 4.3 and the formula 4.1 degree. The word rule base finally increases the decision-making basis of the conditional attributes of the rule base through attribute importance.

4.3 Testing

After the above processing of the statement of the whole training corpus, each word can obtain a rule base with attribute importance, which can be used to optimize the labeling result of the test statement. When a test statement and its NLP preprocessing results are given, the dplist corresponding to each word can be converted into a vector based on the dependent attribute set. Using the cosine similarity method of the vector, one or

more rules corresponding to the maximum similarity $MaxsimSet(dpl)$ can be obtained, and these rules are used to obtain the corresponding decision tag $MaxsimSet(t)$. If a new word (a word that does not appear in the training corpus) appears, temporarily store its tag set in "UNK" , and finally mark the most frequently appearing in $MaxsimSet(t)$ as the word pre-mark(not the final mark).

Given the word segmentation of the test statement and its corresponding pre-marks, the pre-marked result is further analyzed by the algorithm 4.4 to obtain the final mark, and the knowledge subject analysis result is obtained according to the final mark.

Algorithm 4.4 knowledge subject analysis

Input: word segments and its premarks

Output: knowledge subject

1: BLeft = preTagsProcess(tags,'BL')

2: BRight = preTagsProcess(tags,'BR')

3: cut to get the final knowledge subject part by BLeft and BRight

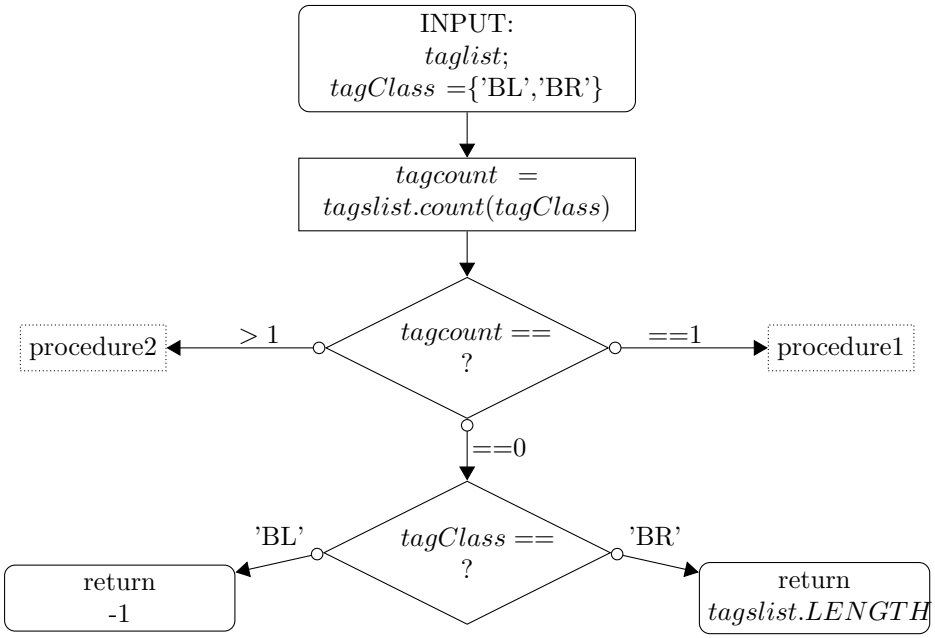

Figure 4.1: preTagsProcess

$$f_{bl}(ti) =$$
$$(ti == 0 \quad or \quad tagslist[ti-1] ==' Neg') \quad and \quad tagslist[ti+1] ==' KSubj' \tag{4.2}$$

$$f_{br}(ti) =$$
$$(ti == len(tagslist) - 1 \quad or \quad tagslist[ti+1] ==' Neg') \quad and \quad tagslist[ti-1] ==' KSubj' \tag{4.3}$$

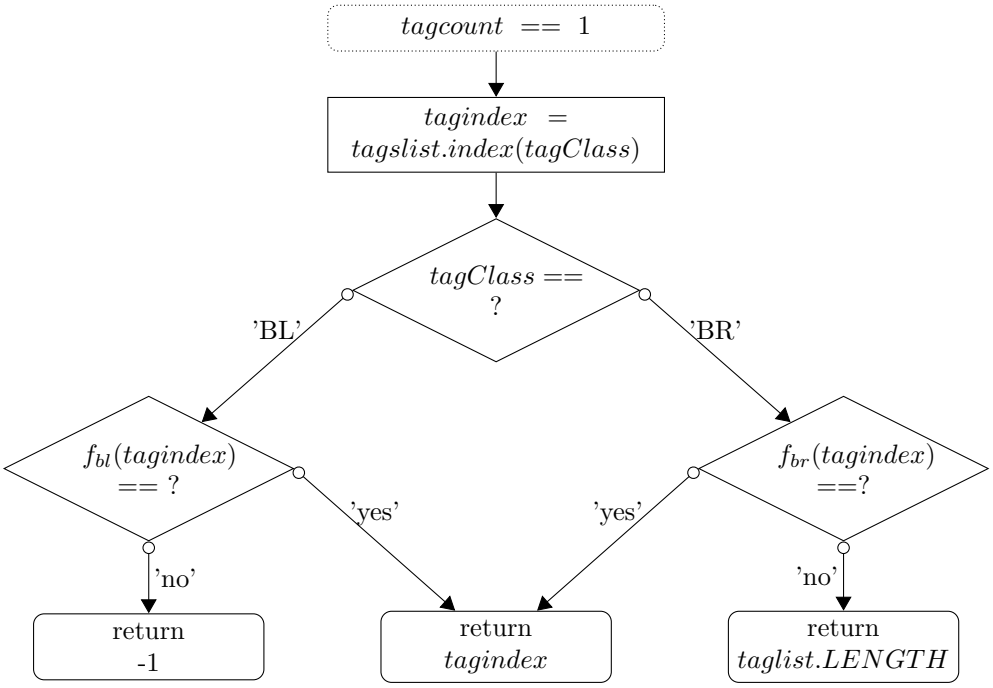

Figure 4.2: procedure1

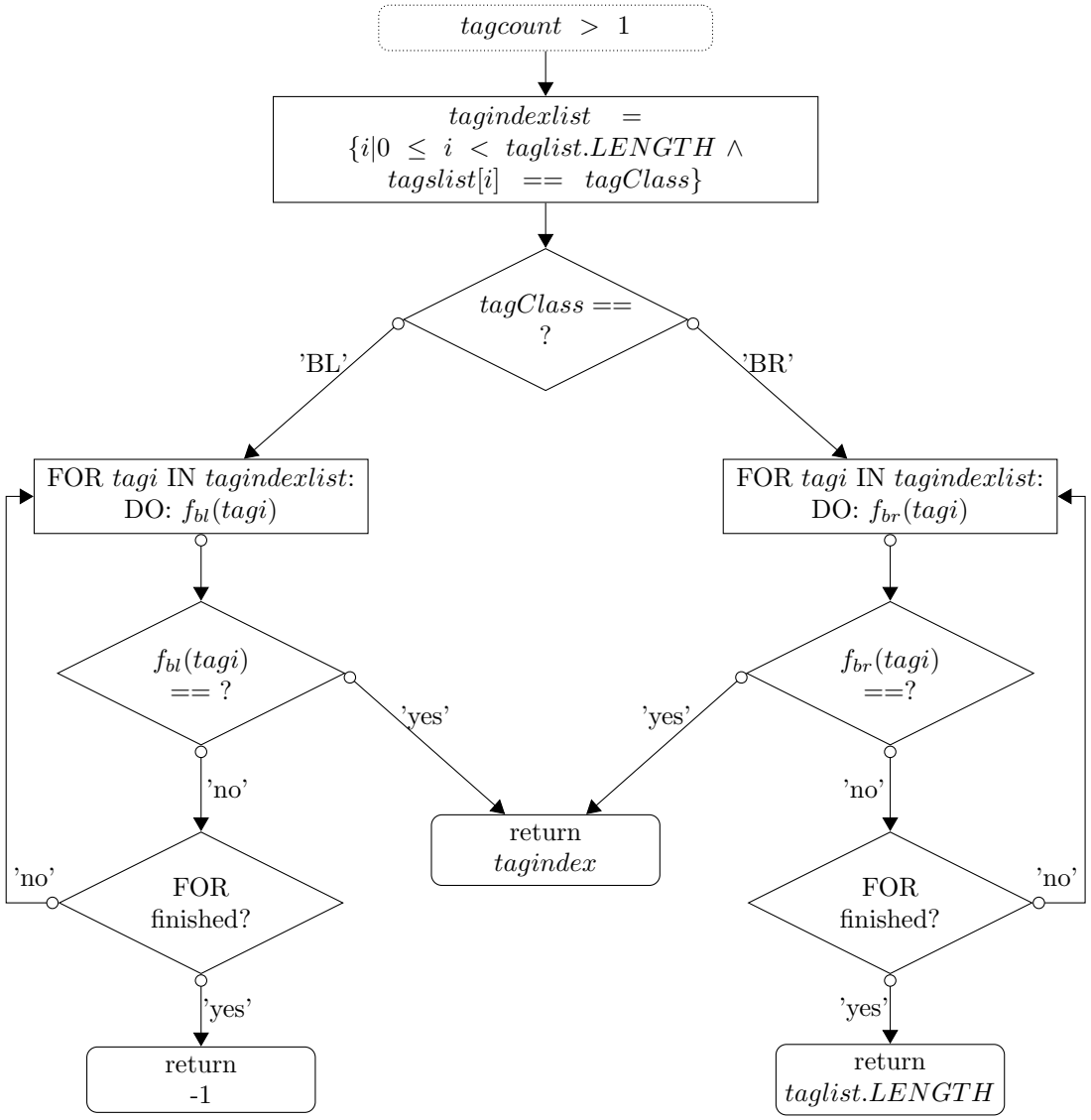

Figure 4.3: procedure2

4.4 Experiment

The experiment used KBQA subtask corpus in the NLPCC2016 evaluation competition after the labeling process by Huang and other people [126, 127, 27], where the training set includes 14609 questions and their corresponding knowledge subject, test set includes 9870

questions and their corresponding knowledge subjects. The StanfordCoreNLP toolkit is used in the dependency analysis process and part of the baseline experiment process. The language model used is the official Chinese processing model. Use the evaluation interface in the Scikit-learn[128] toolkit for results evaluation.

During the experiment, three baseline methods and two related methods in this chapter were compared. The baseline method is CoreNLP's direct syntactic subject method (denoted as StanSubj), CoreNLP's direct named entity extraction method (denoted as StanNER), and the Bi-LSTM-CRF network structure based knowledge subject marking method (denoted as Deep). The two methods in this chapter are: a separate method in this chapter (denoted as Rough) and a method based on the method of this paper to optimize the output of the Bi-LSTM-CRF network (denoted as Deep+Rough). The optimization method is to take the intersection of the results from two methods.

In terms of evaluation indicators, in addition to traditional evaluation indicators (such as accuracy precision, recall rate, F1 value, etc.), a new evaluation indicator is added, as shown in the formula 4.4.

$$newEva(predict, golden) = \frac{NearlyCorrectC(predict, golden)}{C(golden)} \tag{4.4}$$

Among them, *predict* is the set of knowledge subject extracted by experiment, *golden* is the correct set of knowledge subject, $C(golden)$ is the size of the correct knowledge subject set , and $NearlyCorrectC$ is the number of nearly correct subject of the extracted knowledge subject. that is, the formula is used to evaluate the ratio of the "nearly correct knowledge subject" to the total number of correct knowledge subjects. The nearly correct knowledge subject is judged as shown in the formula 4.5.

$$\begin{aligned} &NearlyCorrect(p, g, minSame) \\ &= \begin{cases} True, & \text{if } sameC(p, g) > minSame \text{ and } len(p) <= len(g) \\ False, & \text{the other} \end{cases} \end{aligned} \tag{4.5}$$

Where p is the predictive knowledge subject of a sentence, g is the correct knowledge

subject, and $sameC(p, g)$ is the number of identical Chinese characters between p and g (not the number of participles) , $len(p)$ and $len(g)$ are the number of Chinese characters for predicting the subject of knowledge and the correct subject of knowledge. The decision condition for adding $len(p) <= len(g)$ is to avoid the prediction that the subject length of the knowledge exceeds the correct subject of knowledge and the completely wrong subject of knowledge is also treated as the correct subject of knowledge. It can be seen that the evaluation index gives some fault tolerance to the incomplete extraction of the knowledge subject.

$minSame = 0$ is used during the evaluation of this experiment. The experimental results are shown in the table 4.4.

Table 4.4: experiment results

	StanSubj	StanNER	Deep	Rough	Deep+Rough
newEva	0.2788	0.4258	0.8592	0.3293	**0.8773**
Micro-Precision	0.0536	0.0928	0.7436	0.3182	**0.7599**
Micro-Recall	0.0536	0.0928	0.7436	0.3182	**0.7599**
Micro-F1	0.0536	0.0928	0.7436	0.3182	**0.7599**
Macro-Precision	0.0452	0.0747	0.5918	0.1820	**0.6159**
Macro-Recall	0.0416	0.0729	0.5793	0.1728	**0.6037**
Macro-F1	0.0426	0.0729	0.5834	0.1758	**0.6077**
Avg-Precision	0.06	0.10	0.78	0.35	**0.79**
Avg-Recall	0.05	0.09	0.74	0.32	**0.76**
Avg-F1	0.06	0.09	0.76	0.33	**0.77**

The experimental results show that the single Rough method is obviously superior to the direct method of syntactic subject or named entity as the subject of knowledge. The separate Deep method is superior to the Rough method and the first two methods. However, if the result of the Rough method is used as the output optimization of the Deep method, the combination of the two methods is higher than other methods in each evaluation index.

4.5 Conclusion

This chapter proposes a method for extracting knowledge subject in questions based on attribute importance. Combining the existing method of named entity recognition and syntactic subject analysis, from the perspective of rough set model and rough set attribute importance. Starting, optimize the sequence labeling results for knowledge subject extraction, and improve the overall knowledge subject analysis ability of the system. The experimental results verify the effectiveness of the new method.

The following methods will be optimized from the following aspects: 1) differential sample analysis, further analysis of the reasons for the difference with other methods, so as to find the optimization direction; 2) on the basis of the dependency syntax tuple, add other training features, optimize the effect of the overall characteristics; 3) continue to collect or construct Chinese QA research corpus, and test the method in this way.

Chapter 5

Predicate Analysis for Knowledge Triple

5.1 Introduction

The knowledge-based QA system analyzes the problem from both the knowledge subject and the knowledge predicate. The knowledge subject corresponds to the Resource item in the RDF knowledge tuple, and the knowledge predicate corresponds to the Properties item in the RDF knowledge tuple. The complete knowledge tuple matching must be that the knowledge subject and the knowledge predicate at the same time achieve the highest match value. Whether the correct RDF entry will be matched will affect the accuracy of the resulting answer. Therefore, the knowledge predicate matching of question questions is one of the important links in question analysis.

Due to the uncertainty of knowledge expression and the uncertainty of semantic expression in Chinese language, Chinese knowledge predicate expression is very rich. On the one hand, knowledge predicates may not be directly constructed directly using the participles in the sentence. Under different expressions, the knowledge predicate expressions in the question and the knowledge tuple predicate in the knowledge base are not necessarily identical sets of words. Therefore, it is not possible to extract knowledge predicates directly by syntactic analysis, but must assist the corresponding mapping rules.

On the other hand, even if the expression of the knowledge predicate of the question can be expressed by the word segmentation in the sentence, the knowledge predicate of the

Chinese short question sentence in the question and answer system is difficult to obtain directly by the method of grammar analysis. Similar to the deviation of knowledge subject and syntactic subject, knowledge predicate and syntactic predicate also have deviations, and this situation is more common than knowledge subject deviation.

In addition, due to the flexible expression of Chinese expressions and the meaning of Chinese characters, Chinese sentence subjects and knowledge subjects often have a large number of omitted expressions of Chinese characters and words, which sometimes leads to the lack of obvious syntactic features or predicate features of knowledge predicates in the grammatical analysis results of Chinese sentences.

The above three points are the main difficulties in the analysis of knowledge predicates in Chinese questions. Therefore, how to accurately match differently expressed questions to the corresponding knowledge predicates is a problem that must be solved in the process of knowledge element ancestor matching.

Aiming at this problem, this chapter gives a definition of the knowledge predicate analysis problem for KBQA system from the perspective of rough set theory. On the basis of this definition, a knowledge predicate analysis method based on rough set attribute reduction theory is proposed. The Chinese word is regarded as the attribute of knowledge expression, and the attribute reduction method and boundary domain of rough set are used to mine out attribute words strongly associated with knowledge expression from a given annotated question-knowledge tuple corpus, and enhance the weight of strong association attribute words in subsequent matching by reducing attribute words of weakly associated attributes, and then improves system performance.

5.2 Training

Given a training corpus of a knowledge-based question answering system, including a number of questions and their corresponding knowledge tuples, each question is represented by *ques*, and the knowledge tuple corresponding to the question is represented by KT. The training corpus is segmented and the common punctuation marks such as the question mark in the question are removed to remove the influence of the higher frequency punctuation marks such as the question mark on the conceptual analysis. For each pair

of questions in the corpus - the pairing of knowledge tuples, according to the word set of *ques* and its KT, find the boundary field word set of each knowledge predicate, the process is as follows:

$$\underline{S} = S \cap ques \tag{5.1}$$

$$\underline{P} = P \cap ques \tag{5.2}$$

$$bn(P) = que - \underline{S} - \underline{P} \tag{5.3}$$

The $bn(P)$ word set obtained at this time mainly consists of three parts: 1) boundary domain words that construct the conceptual semantics of knowledge predicates; 2) semantic boundary domain words that construct the concept of knowledge subject; 3) syntactic words needed to construct complete sentences.

When the expression vocabulary of the predicate concept used in the question overlaps with the expression vocabulary in the knowledge tuple, the partial word elements of the boundary domain word set form the same concept as the knowledge predicate with a certain grammar. In the process of knowledge predicate analysis, if there are a large number of overlapping parts of sentence expression and knowledge predicate expression, existing methods based on syntax analysis or similarity matching can map questions to knowledge predicates; if sentence expressions and knowledge predicates If the expression has no overlapping parts or there are few overlapping parts, it may be necessary to use the boundary field word and the predicate decision rule to implement the link between the question and the knowledge predicate.

If the vocabulary that is not related to the conceptual expression of the knowledge predicate is reduced, the remaining boundary domain words will be closer to the syntactic expression of the knowledge predicate, but the overly streamlined reduction results may result in identification difficult of the expression of diverse sentences. For example, if multiple expressions of the predicate concept of ”birthday” appear multiple times in the original training corpus, while other predicate concepts have fewer training examples, the

result of excessive reduction may result in only ”birthday” in the training corpus can be distinguished from the expression of other knowledge predicates, but when other expressions of ”birthday” appear in the test corpus, the expression fails because the expression is reduced. Therefore, the attribute reduction principle of the decision system for knowledge predicate rules is to retain as many words as possible with higher semantic expression and to reduce the words that are weakly related to semantic expression.

Using the training matching of each question-knowledge tuple to do the processing of the previous, the training matching pairs of each question-knowledge tuple can be converted into a matching pair of knowledge predicate-boundary domain word sets. First construct a knowledge predicate decision table $DT = \{U, A = C \cup D, V, f\}$, and the boundary domain word-knowledge predicate obtained after the above preprocessing. The rule base is represented by U, the boundary domain word is the condition attribute C, and the correct knowledge predicate complete expression is the decision attribute $D = \{D_1, D_2, ..., D_i, ...D_n\}$. The attribute value field $V = \{0, 1\}$, 0 means that the word does not exist in the sentence, and 1 means it exists. U/D is used to represent the set obtained by the equivalence division of the decision attribute set D on U. $U_i \in U/D$ is a set of elements marked as i in the set, and the corresponding decision is D_i. $u \in U_i \rightarrow D_i$ is a rule on the decision table, $[u]$ is the precondition of the rule.

In a certain decision equivalence division of the knowledge predicate decision system, if an element appears repeatedly in the rule's precondition, there may be two cases: 1) the word is an important element of the knowledge predicate meaning of the target decision; 2) The word is an important element that constitutes the expression of a question, and is one of the constituent elements of a certain question expression. Therefore, it is an important step to divide the semantic predicate expressions and the semantic expressions of the questions as much as possible. According to the existing rough set knowledge and decision table theory, combined with the rough set reduction theory and decision coverage definition, the following definitions of the rule front analysis table are proposed.

Definition 5.1 *Given* $DT = \{U, A = C \cup D, V, f\}, U_i \in U/D, D = \{D_1, D_2, ..., D_i, ...D_n\}$, *we can get a* $DT' = \{U^*, A = C \cup D, V, f\}$, *in which :*

$$u_i^* = \cup\{a | a \in C, f(a, u \in U_i) = 1\} \tag{5.4}$$

$$U^* = \cup\{u_i^* | U_i \in U/D\} \tag{5.5}$$

According to the definition of 5.1, the rule precondition analysis table is a collection of the boundary domain words of the same concept in the training corpus, and a new decision table is obtained after the collection operation. The main principle of this process is to temporarily reduce the coverage of the knowledge predicate elements in the rule's precondition elements by merging the preconditions of the equivalent decision rules, so that the conditional attribute words that are closer to the meaning of the knowledge predicate are in the overall decision table. The frequency of the sentence is reduced but still retained; while the knowledge predicate expresses the conditional attribute words that are weakly related but strongly related to the semantic expression of the question, although the word frequency is also reduced in the corresponding decision equivalence division, but because it is closer to the question expression, therefore, remains high in the decision table transformed by the question, and the coverage in the overall decision table is still high, thus realizing the knowledge predicate element and the question semantic element separation in frequency.

According to the reduction theory, if the attribute word $a \in C$ exists in U^* of DT' at this time, it satisfies:

$$\frac{\sum_{U^*} f(a, u^*)}{|U^*|} \geq \beta, \quad 0 \leq \beta \leq 1 \tag{5.6}$$

Then the attribute word a can be reduced. The principle of this step is that under the premise that the strong related words have been down-converted, or without repeating the results of the knowledge predicate decision, if the frequency of a word is too high, it means that the word's decision-making ability for knowledge predicate is not high. By reducing the redundant words, we can obtain a set of words that are closer to the expression of the knowledge predicate, which can be used for the judgment of the knowledge predicate.

Then the training process of the knowledge predicate rule system can be obtained, as shown by the algorithm 5.1.

Algorithm 5.1 training for knowledge predicate rule system

Input: training corpus

Output: rule system

1: For training corpus segmentation, for each question-knowledge predicate match pair, use the formula 5.1~expression5.3 to calculate the corresponding $bn(P)$;

2: build knowledge predicate decision table DT;

3: Constructs a rule precondition analysis table DT' using the definition 5.1;

4: According to the formula 5.6, use the traditional frequent item mining algorithm such as Apriori algorithm to reduce the frequent-1 items in the decision table after the collection process;

5: Returns the reduced DT' as a new knowledge predicate decision rule system.

5.3 Testing

After the above processing of the training statement, a library of expression rules of knowledge predicates can be obtained. When the conceptual expression of the knowledge predicate in the sentence has no lexical overlap with the predicate itself, these rule bases can be used to identify the knowledge concept expressed in the sentence. In addition, part of the knowledge predicate itself is the composition of its conceptual expression. Therefore, the test sentence after the word segmentation will perform two similarity calculations: the first time the similarity calculation is performed with all the predicate or predicate phrases, and the item with the highest similarity is returned; the second time is similar to all the rule phrases obtained by the training. The degree calculation returns the predicate corresponding to the item with the highest similarity. The similarity calculation method adopts the cosine similarity method of the TF $\cdot$ IDF vector, and the predicate corresponding to the highest value calculated by the two similarities is the analysis result.

However, in different question and answer systems, due to the different language expression styles and the focus on the field, the knowledge predicate in the question expres-

sion and the knowledge predicate overlap ratio in the knowledge base are also different, so two similarities are needed. The scaling is performed by multiplying a similarity result by a weight adjustment factor of α before performing the comparison process.

In summary, the steps of the knowledge predicate analysis algorithm are shown in the algorithm 5.2.

Algorithm 5.2 knowledge predicate analysis algorithm

Input: test statement, knowledge predicate decision rule system DT', weight adjustment factor α

Output: test statement knowledge predicate

1: Segments the test statement to get the word segmentation sequence and uses TF · IDF vector space to represent it;

2: traverses the predicate library, using the cosine similarity method to calculate the similarity between the test statement and each predicate/predicate phrase, and saves the largest similarity sim_1 and its corresponding predicate term kp_1;

3: traverses the rule base, uses the cosine similarity method to calculate the similarity between the test statement and each rule phrase, and saves the maximum similarity sim_2 and the corresponding predicate term represented by the rule phrase kp_2;

4: Weights the sim_1 value, ie $sim_1 = sim_1 * \alpha$;

5: compares sim_2 with the new sim_1, if sim_1 sim_2, returns kp_1, otherwise returns kp_2. The return item is the knowledge predicate sought.

5.4 Experiment

The experiment used KBQA subtask corpus in the NLPCC2016 evaluation competition after the labeling process by Huang and other people [126, 127, 27], The training set includes 14609 questions and their corresponding knowledge predicates, and the test set includes 9870 questions and their corresponding knowledge predicates. The results of the evaluation interface in the Scikit-learn[128] toolkit were used to evaluate the results. The word segmentation uses the python interface version PYNLPIR of ICTCLAS[125].

The experiment is divided into two parts: the comparison experiment and the performance observation experiment. The specific implementation plan is as follows:

In the comparative experiment, this chapter uses three baseline methods for comparison. Method 1 is to directly match the similarity of the question to the predicate lexicon (denoted as predCmp); the method 2 is to match the question and the predicate lexicon with the original training question, and the question is regarded as the predicate expression without complete reduction(denoted as QtpCmp); Method 3 is to obtain the decision analysis table DT' according to the initial decision table but not to perform reduction, and then perform similarity matching operation, and no weight adjustment is performed during the matching process (denoted as preRough), Method 4 uses the complete algorithm 5.1 and the algorithm 5.2 step to obtain the experimental results (denoted as fullRough). The reduction step in the training process uses the Apriori algorithm, and the filter threshold β is set to 0.75. The text vector space representation of tfidf is used for all texts, and the matching operation uses the cosine similarity of the vector to calculate the score.

In order to observe more obvious performance change effects, the experimental results of macro-average and micro-average correlation are displayed in groups by using the two-axis method, with Precision, Recall, and F1 as categories. Since the numerical results of the comprehensive average performance experiment are less fluctuating and the trend of change is more obvious, only a single axis is used.

In the comparison experiment, when the weight adjustment coefficient α is 1.4, the experimental results are shown in the table 5.1.

Table 5.1: results of $\alpha = 1.4$

	predCmp	QtpCmp	preRough	fullRough
Macro-precision	0.5974	0.5110	0.5889	0.5891
Micro-precision	0.5134	0.4593	**0.5464**	**0.5468**
Macro-recall	0.5106	0.4582	**0.5444**	**0.5444**
Micro-recall	0.5134	0.4593	**0.5464**	**0.5468**
Macro-F1	0.5168	0.4536	**0.5336**	**0.5306**
Micro-F1	0.5134	0.4593	**0.5464**	**0.5468**
Avg-Precision	0.61	0.52	0.59	0.60
Avg-Recall	0.51	0.46	**0.55**	**0.55**
Avg-F1	0.52	0.46	**0.54**	**0.54**

It can be seen that the preRough method and the fullRough method are superior to the predCmp method and the QtpCmp method on most indicators. In the two indicators of Average Precision and Macro-precision, the value of the evaluation index of the preRough method and the fullRough method is slightly lower than that of the predCmp method. The expression of knowledge predicates in the questions in the test corpus is relatively straightforward, so there are more identical parts of the question and knowledge predicates. In practical applications, the preRough method and the fullRough method can handle more linguistic expressions.

In the performance analysis experiment, the experimental results are shown in the figure 5.1~5.8.

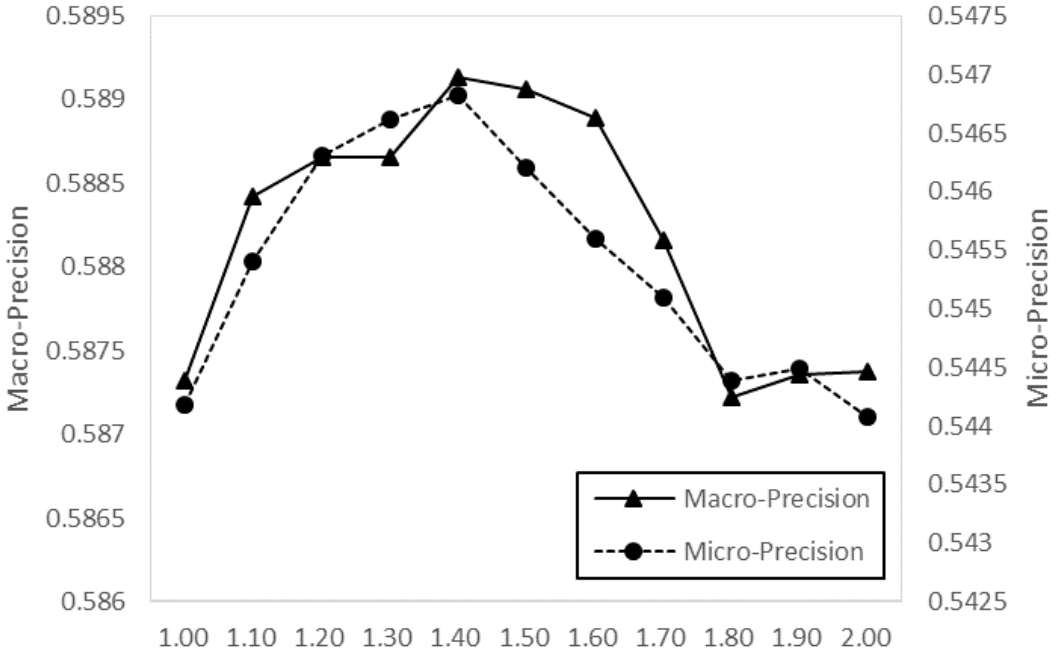

Figure 5.1: [1,2],step = 0.1(precision)

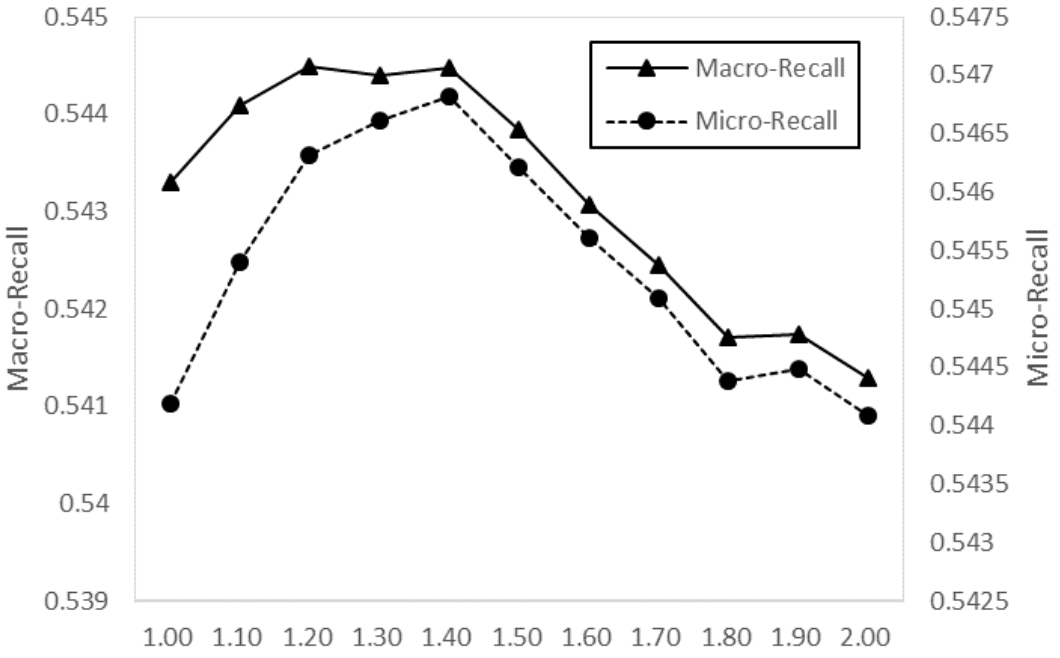

Figure 5.2: [1,2],step = 0.1(recall)

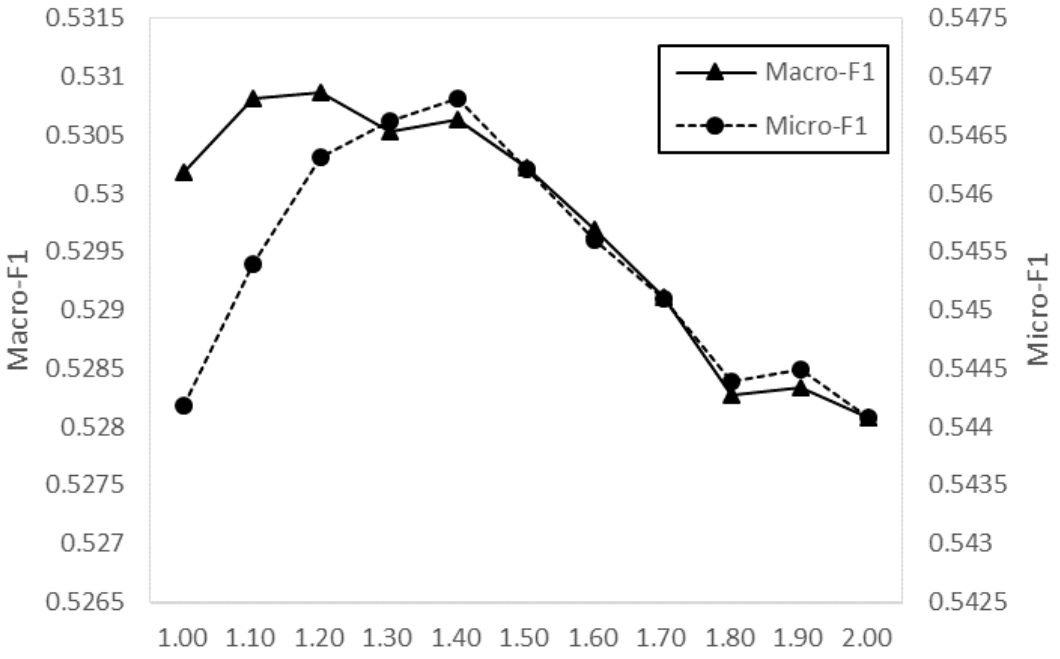

Figure 5.3: [1,2],step = 0.1(F1)

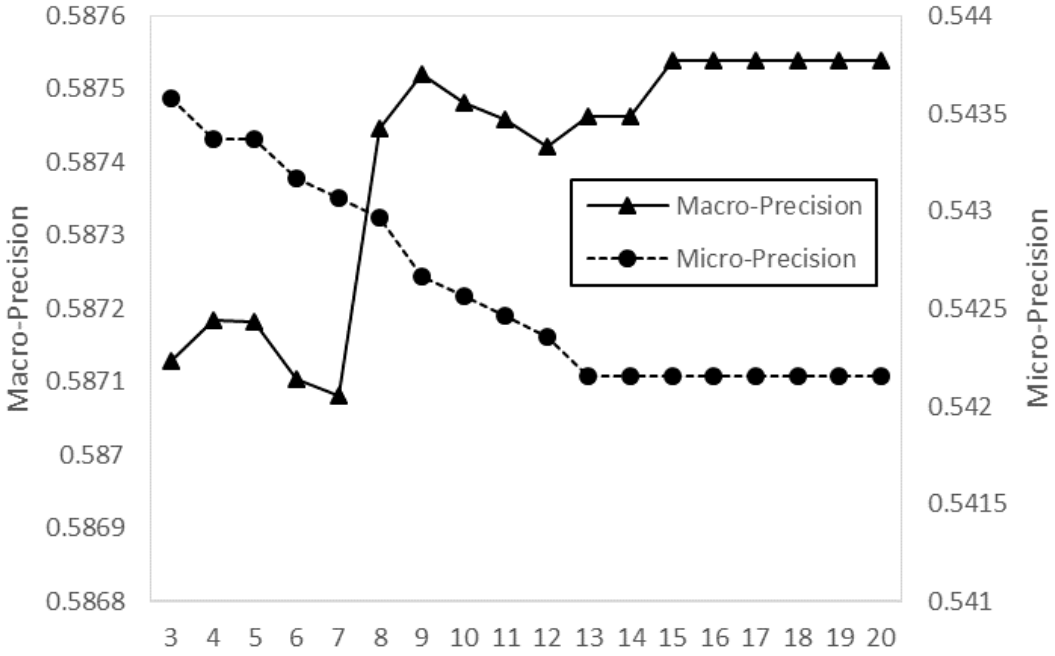

Figure 5.4: [3,20],step = 1(precision)

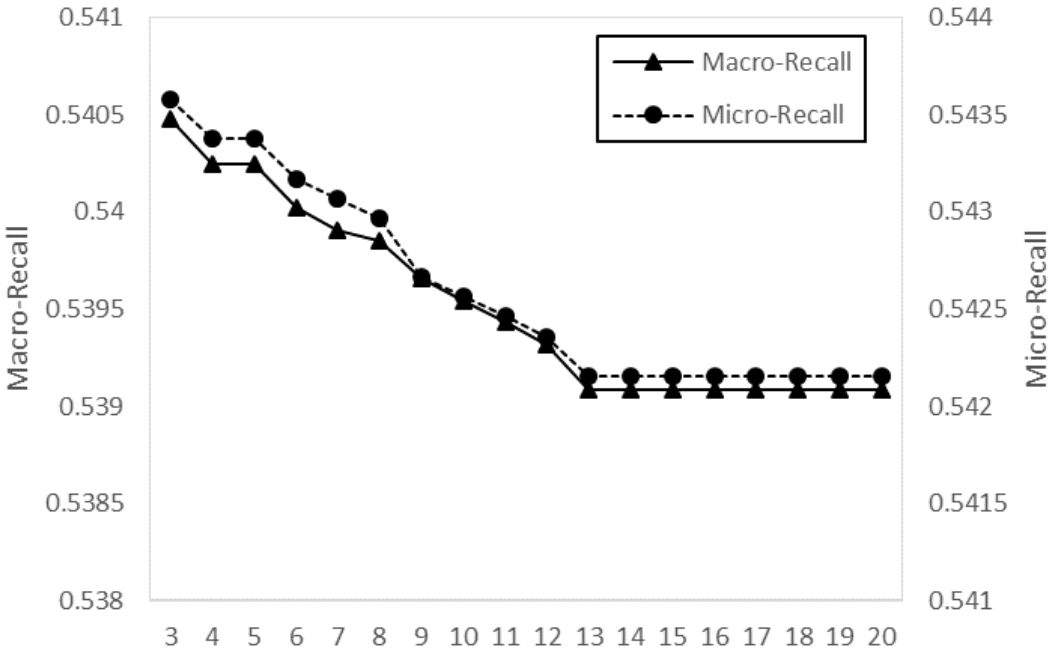

Figure 5.5: [3,20],step = 1(recall)

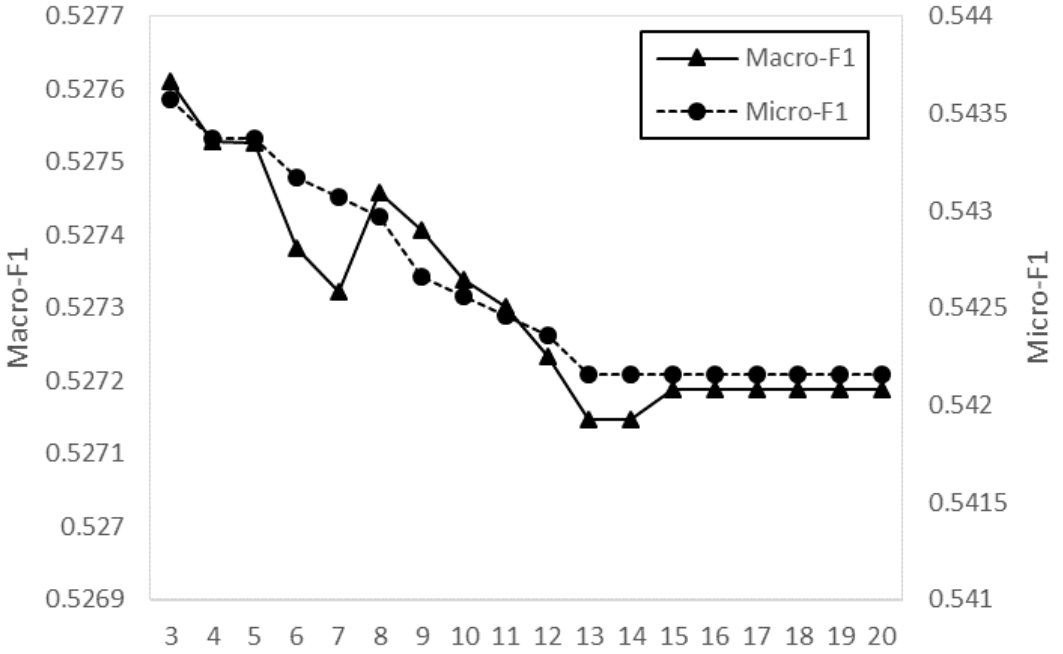

Figure 5.6: [3,20],step = 1(F1)

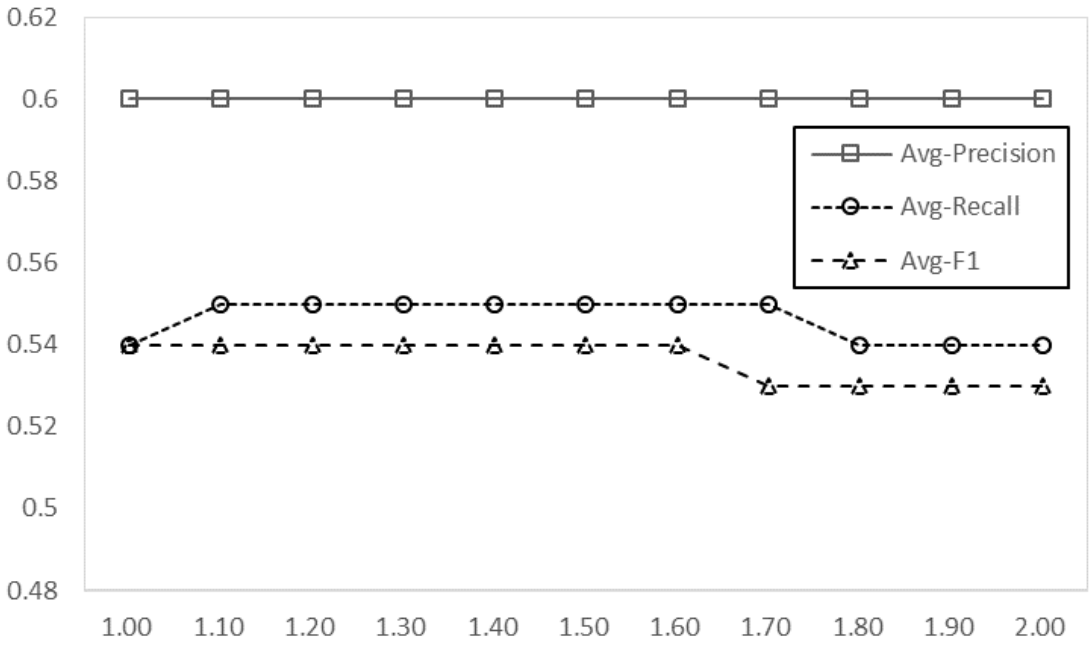

Figure 5.7: [1,2],step = 0.1(overall)

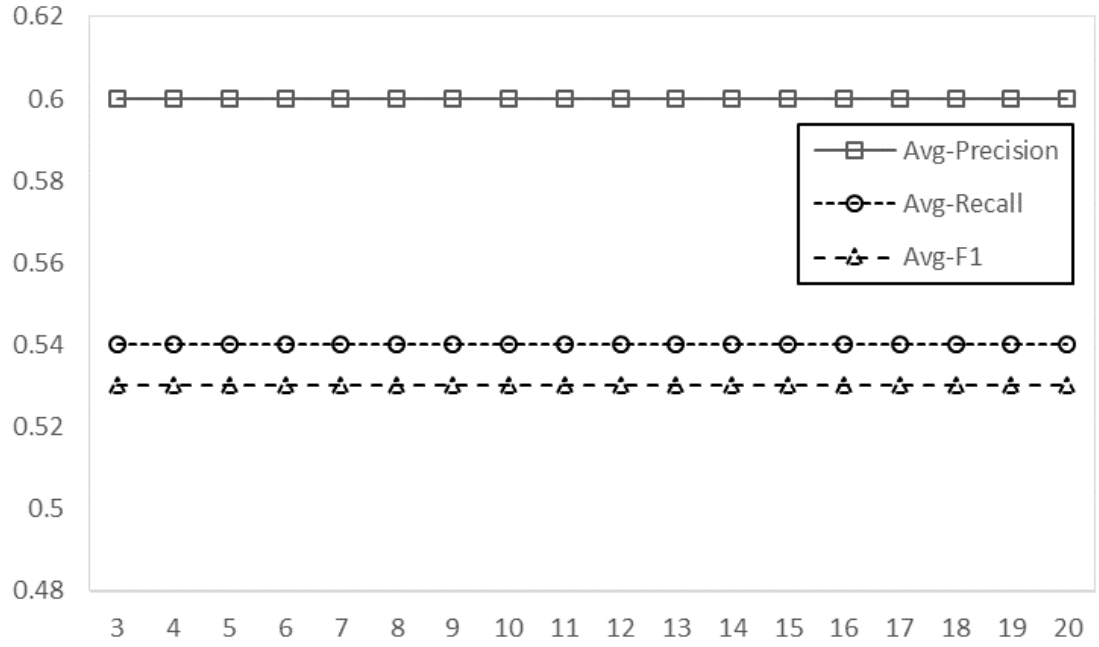

Figure 5.8: [3,20],step = 1(overall)

Figure 5.1 ~Figure 5.3 shows that the value of Precision, Recall, and F1 reaches the maximum value when $\alpha = 1.4$ (results are shown in the table 5.1). When $\alpha < 1.4$, the three types of indicators generally show an upward trend, and after $\alpha > 1.4$, the overall trend shows a downward trend. This is because when the input question in the system is more inclined to express with a higher expression than the knowledge predicate, if α is too small, the result of direct matching of the knowledge predicate cannot exceed the knowledge predicate rule with more relative expression elements; if α exceeds a certain value, the final adjusted sim_1 value is too high, the predicate rule does not work.

Figure 5.4 shows that after $\alpha > 3$, the precision of the Precision class will fluctuate, but the highest peak still does not exceed the performance of $\alpha = 1.4$; chap05:fig:3-20recall ~Figure 5.6 shows that in the $[3, 20]$ interval, the performance of the Recall and F1 classes generally showed a downward trend, and the peak value was not over performance at $\alpha = 1.4$. Figure 5.7 and Figure 5.8 show that α reaches the most in the range of $[1.1, 1.6]$ for Avg-Recall and Avg-F1 Good performance, after $\alpha \geq 3$, all the comprehensive performance will not change.

In summary, in the corpus used in this experiment, the value of α is set to about 1.4 to achieve the best performance of the system.

5.5 Conclusion

From the perspective of rough set theory, this chapter presents a new way of analyzing the rules of knowledge predicate in Chinese knowledge question answering system, and proposes a method of analyzing question predicate based on rough set theory. Compared with the traditional method, this method can better recognize the expression of diverse knowledge predicates on the basis of ensuring performance. The experimental results verify the effectiveness of the method.

The advantages of the method proposed in this chapter are mainly two aspects: First, when there are enough training examples, the method can store a variety of knowledge predicate expressions; if the adjustment coefficient α is set enough, it can be output from the rule match. The result is optimally selected from the output results directly matched by the knowledge predicate list. However, there are still some problems in the methods in this chapter, such as the combination or evaluation of synonym knowledge predicates, the output selection of sim_1 and sim_2, so the subsequent problems still need to be optimized.

Chapter 6

Affective Computing for Short Text

6.1 Introduction

Here we focus on two problems in short text analysis(Here a "short text" is defined as a single sentence ended with a period. It is text different from the long paragraph with more than 2 sentences and sentence relation within the paragraph). One is that, for social dialogue or Twitter, the flexible change of spoken language and Internet language is very common. Different from static document text such as a novel or a report, the new language information can be dynamically added into the existing context, such as by sending comments to a twitter or adding further explanations during a Human-Robot Interaction. Another is that based on the running environment and system resource limits, the system must allow the adjustment of the fineness of the semantic analysis and affective computing. For example during a Human-Robot Interaction, sometimes the robot only needs to classify the user's input into tree sentiment classes: *positive*, *neutral* and *negative*; while sometimes it also needs to discriminate the sentiment changing extent between two sentences of the same class, such as from *common positive* to *extreme positive*. To solve these problems, a language representation model named Synthetic and Computational Language Model(SCLM) was proposed by Zhao Han et al.[103], which represents modifying and modified information using matrices and vectors respectively. In SCLM, the modifying matrices contain more affective information than the vectors, while the modi-

fied vectors contain more semantic information than the matrices. The determinant value of the matrix will be the sentiment tendency value of a modifying word or a sentence part.

6.2 Proposed Method

6.2.1 Sentence Representation and Sentiment Measurement of SCLM

Using SCLM a single declarative sentence can be represented by *Formula(6.1)*:

$$S = \begin{cases} T_0, & n = 0 \\ T_0 + \sum_{i=1}^{n} C_i, & n = 1, 2, 3, ... \end{cases} \tag{6.1}$$

T_0 represents the main trunk clause of the sentence and C_i represents the subordinate clauses. n means the amount of the subordinate clauses.

For T_0 and each C_i, use *Formula(6.2)* to represent:

$$\left(\prod_{j=0}^{m} M_{kj} \right) V_k, \quad k, j, m = 0, 1, 2... \tag{6.2}$$

M_{kj} or V_k is the smallest unit of the model representation. When k is 0, V_k represents the main trunk T_0, and when k is other integer, V_k represents the clause C_i. For each V_k the elements in it represent the modified words in the sentence, and the elements in M_{kj} must be corresponding to those in the modified vector. *Formula(6.3)* with *Formula(6.4)* is an example using a vector of 4 dimension and the modifying matrix is $4 * 4$ size:

$$V_k = \begin{bmatrix} subject \\ predicate\ verb/copula \\ direct \quad object \\ indirect \quad object \end{bmatrix} \tag{6.3}$$

$$
\begin{aligned}
M_{kj} &= \begin{bmatrix} DM_{subj} & s-p & s-d & s-i \\ p-s & DM_{prev} & p-d & p-i \\ d-s & d-p & DM_{dobj} & d-i \\ i-s & i-p & i-d & DM_{iobj} \end{bmatrix} \\
&= \begin{bmatrix} row_{subj} \\ row_{prev} \\ row_{dobj} \\ row_{iobj} \end{bmatrix}
\end{aligned}
\tag{6.4}
$$

In M_{kj}, *"subj"* means *"subject"*, *"prev"* means *"predicate verb or copula"*, *"dobj"* means *"direct object"*, and *"iobj"* means *"indirect object"*. The elements on matrix diagonal such as DM_{subj} are the most direct elements modifying the vectors(Directly Modifying). row_{subj}, row_{prev}, row_{dobj}, row_{iobj} represent the rows of the matrix. Elements at the other positions show the hidden relations between the indirect modifying words and the modified words. Because of the multiplying rules between a matrix and a vector, the element *"A-B"*, different from the element *"B-A"*, has a directional meaning from A to B. For example, *"s-p"* means the effects from the *"subject"* to the *"predicate verb/copula"*, and will function on the *"predicate verb/copula"* part of the vector after the multiplying process, while *"p-s"* means the effects from the *"predicate verb/copula"* to the *"subject"* and will function on the *"subject"*. If the element on the position of *"A-B"* is empty, it means that there is no modifying relationship from *"A"* to *"B"*.

For each M_{kj}, $|M_{kj}|$ which represents the determinant value of M_{kj}, can be used as a kind of sentiment value in the sentence. The determinant can be calculated using the determinant calculation method in Linear Algebra[129][130]. Since the matrix rows have corresponding modifying relationships with the vector elements, we can use the vector like the one described in *Formula(6.3)* to represent the selected syntax features. So we can say the selected syntax features of the given 4 dimension SCLM are *"subject"*, *"predicate verb/copula"*, *"direct object"* and *"indirect object"*. If we get all the syntax features, the

elements not on the diagonal can be deduced by the diagonal elements.

Here is a simplified example of decoding a sentence: *"The film is painfully authentic, and the performances of the young players are utterly convincing."*, using a vector of 4 dimension and matrices of the 4x4 size (For a clear formula expression we omit the elements that are not on the matrix diagonal, and use *"□"* to represent an empty position of the diagonal), like *Formula(6.5)*. This sentence is selected from the Stanford Treebank Dataset[131].

$$
\begin{aligned}
S &= T_0 + C_1 \\
&= (M_{00})\, V_0 + \left(\prod_{j=0}^{1} M_{1j} \right) V_1 \\
&= \begin{bmatrix} The,\ painfully\ authentic & & & \\ & (is) & & \\ & & \square & \\ & & & \square \end{bmatrix} \begin{bmatrix} film \\ be \\ \square \\ \square \end{bmatrix} \\
&\quad + O(and) \\
&\quad \begin{bmatrix} the,\ (s),\ of\ the\ young\ players,\ utterly\ convincing & & \\ & (are) & \\ & & \square \quad \square \end{bmatrix} \\
&\quad \begin{bmatrix} performance \\ be \\ \square \\ \square \end{bmatrix}
\end{aligned}
\tag{6.5}
$$

6.2.2 Sentiment Measurement in Proposed Method

The idealized SCLM is very difficult to realize. One of the problems is that there is still no effective definition for the operation of SCLM, such as matrice multiplying in C_1 of *Formula(6.5)*. Another problem is that there is still no effective definition for us to get the modifying value of hidden relation. For example, in the sentence *"The performances of the young players are utterly convincing. "*, we can get the modifying relation from *"utterly"* to *"convincing"*, visually or based on the parsing tree[131], while it is difficult for us to get the

modifying value from *"utterly"* to *"performance"*. In addition, SCLM depends too much on the dependency parsing results, but recently there is still no effective tool for parsing a sentence with complex semantic dependency relations into SCLM representation. So in practical applications we usually use a simplified SCLM model to measure the sentiment of a sentence. We construct only one matrix and use the determinant value of this matrix as the sentiment value of the sentence. All the elements of the same parsing dependencies will be abstracted and the average value of the sentiment tendencies will be used to calculate the elements value in the modifying matrices. For example, in a sentence, the words which are modifying the direct objects, both in main trunks and subordinate clauses will be abstracted and the average of the sentiment values of them will be used. Then for one sentence with no matter how many clauses it has, we just need to calculate the determinant value of only one matrix. The matrix considers only the modifying words that directly modify vector elements, and uses only the sentiment value on its diagonal.

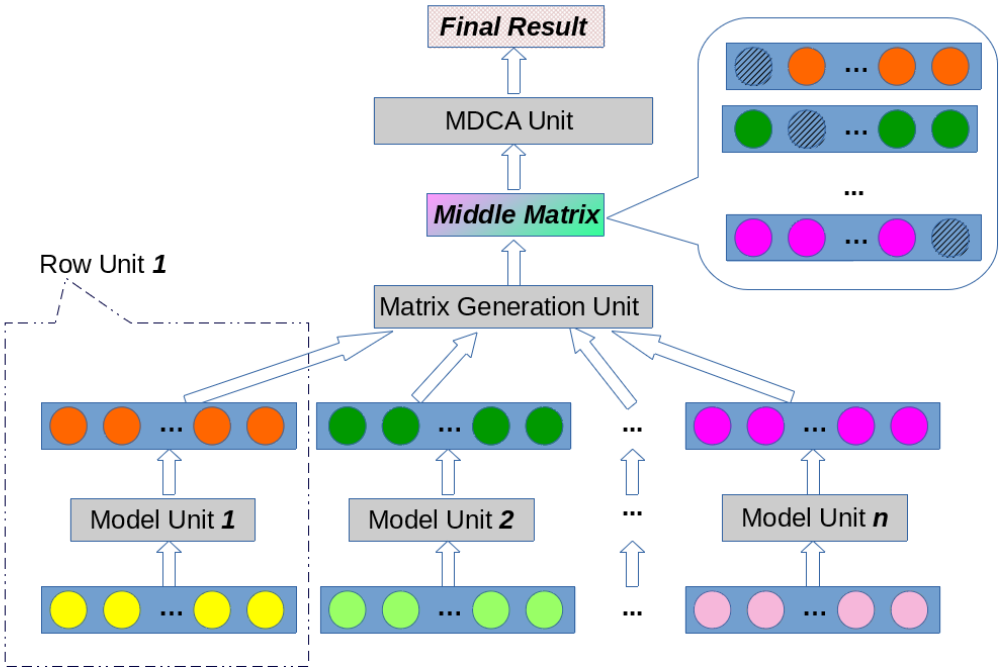

Figure 6.1: Framework of Sentiment Matrix Constructor

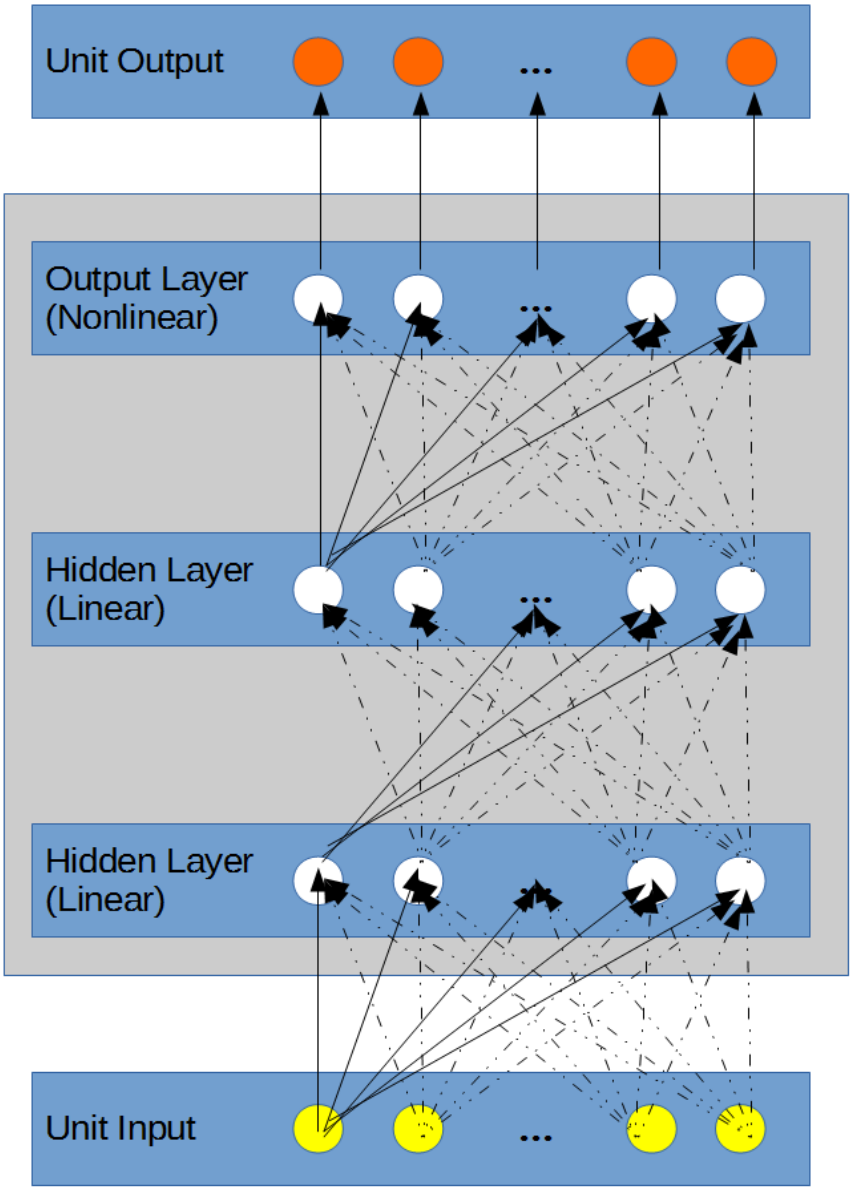

Figure 6.2: Row Unit: with a Model Unit inside

The sentiment measurement of simplified SCLM is very good at discriminating the sentiment changing of two similar sentences, but the sentiment analysis performance of SCLM is unsatisfactory. To solve this problem, we develop a sentiment model based on the SCLM, and focus on the task of sentiment analysis. A sentiment matrix constructor in n dimension is mainly made up by n Row Units, a Matrix Generation Unit and a Matrix Determinant Calculation and Adjustment(MDCA) Unit, illustrated in Fig.6.1. A Row Unit example is illustrated in Fig.6.2, with a Neural Network Model inside used as Model Unit(surrounded by a big gray rectangle), which is composed of several linear layers as hidden layers and a nonlinear layer as output layer. The Model Unit receives the input values of the Row Unit and after model function it will transmit the output values. A set of output values from one Row Unit will be passed to the Matrix Generation Unit as one row of the matrix. The Matrix Generation Unit will construct a sentiment matrix M, and add a model bias b on the diagonal elements of M to get a new matrix M' (Details about b will be introduced in Section(3.3)). The determinant value of M' will be calculated and adjusted by the MDCA Unit and then we will get the final sentiment value of the sentence.

If we choose n syntax features of the sentence, the modifying matrix M of the proposed method will be represented by *Formula(6.6)*:

$$
\begin{aligned}
M &= \begin{bmatrix} m_{11} & m_{12} & ... & m_{1n} \\ m_{21} & m_{22} & ... & m_{2n} \\ ... & ... & ... & ... \\ ... & ... & m_{ij} & ... \\ ... & ... & ... & ... \\ m_{n1} & m_{n2} & ... & m_{nn} \end{bmatrix} \\
&= \begin{bmatrix} row_1 \\ row_2 \\ ... \\ row_i \\ ... \\ row_n \end{bmatrix} \\
&= \begin{bmatrix} f_1(F_1) \\ f_2(F_2) \\ ... \\ f_i(F_i) \\ ... \\ f_n(F_n) \end{bmatrix}, \qquad (i, j = 1, 2, ..., n) \quad .
\end{aligned}
\tag{6.6}
$$

In *Formula(6.6)*, m_{ij} is the matrix element on the row i and the column j, row_i is all the elements on the row i, f_i is the model function of the Model Unit corresponding the row i, F_i is the set of feature values of the row i. The feature values are based on the feature words, and details of the feature words and feature values will be introduced in Experiment Section.

6.2.3 System Training in Proposed Method

The training process is divided into two parts: Row Units training and MDCA Unit training. We must train the Row Units at first and then train the MDCA unit. After we input a training sentence into the system, each Row Unit of the system will be adjusted based on the target sentiment value T of the sentence and the determinant value $|M|$ of the matrix M. The adjustment rate r will be calculated by *Formula(6.7)* :

$$r = \frac{T}{\sqrt[n]{|M|}} \tag{6.7}$$

Since for real numbers, when the n is an even number, the object of the square root operation must be a non-negative number. So we must make the $|M|$ be a positive number. A simple method is that we can add positive bias b on all the elements of the diagonal of the matrix. The b must be large enough to keep the matrix determinant being a positive value all the time. The new matrix with bias represented by M' will be represented by *Formula(6.8)*:

$$\begin{aligned} M' &= M + b * E \\ &= \begin{bmatrix} m_{11}+b & m_{12} & ... & m_{1n} \\ m_{21} & m_{22}+b & ... & m_{2n} \\ ... & ... & ... & ... \\ ... & ... & m_{ij} & ... \\ ... & ... & ... & ... \\ m_{n1} & m_{n2} & ... & m_{nn}+b \end{bmatrix} \end{aligned} \tag{6.8}$$

and the rate r and target sentiment value T should also be changed into *Formula(6.9)* and *Formula(6.10)*:

$$T' \approx T + \boldsymbol{b} \tag{6.9}$$

$$r' = \frac{T'}{\sqrt[n]{|M'|}} \tag{6.10}$$

If we want the determinant value to be positive, the matrix must meet a condition that in each row the element value on the diagonal must be larger than the sum of the absolute value of other elements[132][133][134]. In the system case, if all the elements in Matrix M do not exceed 1, we can set the b with *Formula(6.11)*:

$$\boldsymbol{b} > n - 2 \tag{6.11}$$

After getting the adjustment rate r, the r will adjust all the elements of the matrix, and then the adjusted element values will be passed to the Row Units as training targets. The procedures are illustrated in Fig.6.3. We can treat this step as a process to train a regression Model for matrix M as input and matrix M_t as output (M_t is showed in *Formula(6.12)*). Using Back-Propagation Neural Network Algorithm[115] the weights in the Row Units will be changed.

$$M_t \quad = \quad M' * r' - b * E \tag{6.12}$$

In the Row Unit training, the training rate r is based on $T + \boldsymbol{b}$ and $|M'|$, while in most cases T' doesn't equal with the $T + \boldsymbol{b}$ [130]. So after we train the Row Units, we must train the MDCA Unit to do the adjustment. That means, we train a regression to fit the $\sqrt[n]{|M'|} - b$ generated by the trained Row Units and the Matrix Generation Unit to approach the real sentiment value T.

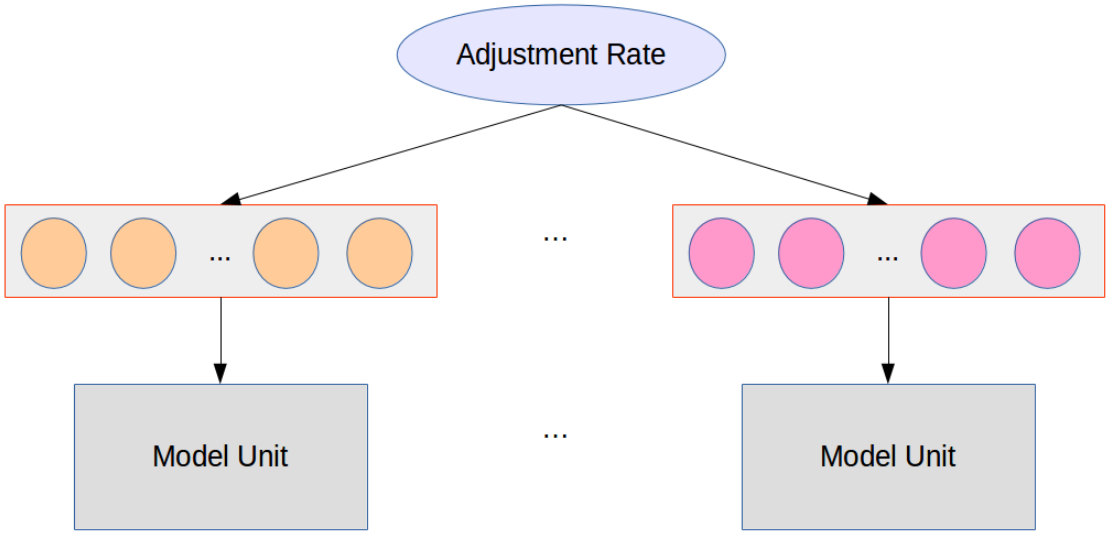

Figure 6.3: System Training

6.3 Experiment

6.3.1 Environment Setting

We use a method which has been used by the research group of Stanford Treebank Dataset[131] to represent the sentiment value. Using the continuous value from 0 to 1, and 0 means the most negative and 1 means the most positive, the sentiment value of a sentence or a word will be represented by the value like a probability. Since the sentences in Stanford Treebank Dataset are all short sentences, we also use the Stanford Treebank Dataset as training corpus. The dataset is divided into 3 parts the same as in reference [118]: of all the 11855 sentences, 8544 sentences are used for training the Row Units; 1101 sentences are used for training the MDCA Unit, and the last 2210 sentences are used for test. The sentiment value on each sentences set is in a uniform distribution.

We use Neural Network(NN) to build and train the Row Units and MDCA Unit,and use PyBrain toolkit to construct and train the NN units[135]. In each NN, the output layer uses the Sigmoid Function[136][137] Layer and the hidden layer uses Linear Function. We train the NN units until convergence but not exceed the max epochs 30. The sentiment matrix is in 3 dimension corresponding modifying *subject*, *direct object* and *indirect object*, and with bias b set to be 3 adding to each element on the diagonal of matrix. For each Row

Unit, the NN layers and nodes are set by $RULayer = [3, 3, 3, 3, 3, true]$, means 1 layer with 3 feature nodes for input, 3 hidden layers with 3 nodes in each hidden layer, and 1 output layers with 3 output nodes, and with $NNBias = True$. The MDCA Unit is set by $MDCALayer = [1, 3, 3, 3, 1, true]$. All the parameters of NNs are initialized by random at the beginning of each training.

We use the dependency parsing module of the Stanford CoreNLP Toolkit [138] to abstract the modifying dependency relations. Each dependency relation is directive from a source word to a target word, for example, the relation *advmod*(adverb modifier) from source word *"convincing"* to target word *"utterly"*. We first get all the Vector Feature Words(VFW): both source and target words of all the subject-relative relations such like *nsubj*(nominal subject), *nsubjpass*(passive nominal subject),etc., and passed to the *subject* row of the vector. Then we get target words of all the direct-object-relative relations and pass them to the *direct object* row, and get target words of all the indirect-object-relative relations and pass them to the *indirect object* row. Then we abstract all the sentiment-relative and modifying-relative relations such as *advmod*(adverb modifier), *amod*(adjectival modifier), *neg*(negation modifier) , etc. and then use the target words as Matrix Feature Words(MFW). The feature words will be passed to the matrix row, vector feature words of which is the neast in the dependency parsing tree. After this step, we can get initial vector feature words and initial matrix feature words from the example sentence. The feature words were shown in *Formula(6.13)* and *Formula(6.14)*, using *"□"* to represent the empty position):

$$
\begin{aligned}
VFW &= \begin{bmatrix} vfw_{subj} \\ vfw_{dobj} \\ vfw_{iobj} \end{bmatrix} \\
&= \begin{bmatrix} film, & performance \\ & \square \\ & \square \end{bmatrix}
\end{aligned}
\tag{6.13}
$$

$$
\begin{aligned}
MFW &= \begin{bmatrix} mfw_{subj} \\ mfw_{dobj} \\ mfw_{iobj} \end{bmatrix} \\
&= \begin{bmatrix} painfully,\ authentic,\ young,\ utterly,\ convincing \\ \square \\ \square \end{bmatrix}
\end{aligned}
\tag{6.14}
$$

Based on the matrix feature words, we can get feature values F_i on each row, see *Formula(6.15)*: the average of sentiment values of the matrix features words on each row $AVG(mfw_i)$, the average of sentiment values of the negative modifying words on each row $AVG(neg_i)$, and the average of the sentiment values of all the words in the sentence $AVG(S)$. The sentiment value of each word and the full sentence can be searched from the sentiment dictionary of Stanford Treebank Dataset[131]. Empty positions of the matrix represented by "□" will be set by 0.5 which means neutral or no sentiment modifying. These matrix feature values will be used as the input of each Row Unit and the sentiment value of the full sentence will be used as the target value T.

$$
\begin{aligned}
F_i &= [AVG(mfw_i), AVG(neg_i), AVG(S)] \quad , \\
&\quad (i = subj, dobj, iobj) \quad .
\end{aligned}
\tag{6.15}
$$

The experiment is divided into three parts: regression, 3-classes classification and 5-classes classification. The regression evaluation is based on Mean Absolute Error(MAE), see the *Formula(6.16)*:

$$
MAE = \frac{1}{n}\sum_{1}^{n}|e_i| = \frac{1}{n}\sum_{1}^{n}|f_i - y_i| \tag{6.16}
$$

e_i means the error between the system output value f_i and the target value y_i. In our

experiment, MAE value is not allowed to exceed 0.2, that means the sentiment value error will not exceed a class distance of 5-classes of the Stanford Treebank: *"very negative"*, *"negative"*, *"neutral"*, *"positive"*, and *"very positive"*.

The classification evaluation is compared with simplified SCLM and the sentiment computing module of Stanford CoreNLP with a trained model on the official net. we compare the performance of 3-classes and 5-classes classification among the three methods. The 3 classes include *"negative"*, *"neutral"*, *"positive"*, with the sentiment value range of [0,0.4), [0.4,0.6), [0.6,1]. The 5 classes include *"very negative"*, *"negative"*, *"neutral"*, *"positive"*, and *"very positive"*, with the sentiment value range of [0,0.2), [0.2,0.4], [0.4,0.6], [0.6,0.8], [0.8,1]. The evaluation of classification is based on precision, recall, F1 value, macro-precision, macro-recall and macro-F1 value, see from *Formula(6.17)* to *Formula(6.22)*:

$$Precision_c = \frac{T_c}{T_c + F_c} \tag{6.17}$$

$$Recall_c = \frac{T_c}{|Set_c|} \tag{6.18}$$

$$F1_c = \frac{2 * Precision_c * Recall_c}{Precision_c + Recall_c} \tag{6.19}$$

$$MacroPrecision = \frac{1}{|C|} \sum_{c \in C} Precision_c \tag{6.20}$$

$$MacroRecall = \frac{1}{|C|}\sum_{c \in C} Recall_c \tag{6.21}$$

$$MacroF1 = \frac{2 * MacroPrecision * MacroRecall}{MacroPrecision + MacroRecall} \tag{6.22}$$

6.3.2 Results and Analysis

We trained 31 models, with all the parameters initialed randomly at the beginning of each training. Each train and test will spend about 5 hours. Then we choose two models: one has best performance in 3-classes classification(represeted as *"Best 3c Model"*) and the other has best performace in 5-classes classification(represeted as *"Best 5c Model"*). 3-classes experiment results are listed in Table 6.1 and 5-classes experiment results are listed in Table 6.2. We also list the 3-classes classification test results of the best 5-classes Model in Table 6.3. The SCLM and the Best 3c Model have no successful results in 5-classes experiment, because the precision and recall values of some sub-classes are 0. So we only compare the 5-classes experiment results only between the CoreNLP and the proposed method. The MAE value of the best 3-classes model is 0.1984, and the MAE value of the best 5-classes model is 0.1787.

The classification results show that the proposed method can deal with 5-classes classification task successfully, and on most evaluation parameters of 3-classes, the proposed method has improved most of the performances than the two method based on simplified SCLM(only the recall of the neutral in 3-classes is not improved). The improvements are showed in Table 6.4. The precision of Best 3c Model has been improved 1.42% $\sim$ 24.52%, the F1 has been improved 1.60% $\sim$ 25.65%. When compared with the Stanford CoreNLP in both 3-classes and 5 classes, there are also some cases that the new method has better performance(the cases have been marked in bold). All the MAE values of these models did not exceed 0.2.

Table 6.1: 3-classes Experiment Results

	SCLM	SCLM(b)	CoreNLP	Best 3c Model
$Precision_{positive}$	0.7273	0.8095	0.7761	**0.8494**
$Precision_{neutral}$	0.1772	0.1772	0.3881	0.1914
$Precision_{negative}$	0.6000	0.6000	0.6612	**0.8452**
$MacroPrecision$	0.5015	0.5289	0.6085	**0.6287**
$Recall_{positive}$	0.0088	0.0187	0.7701	0.1551
$Recall_{neutral}$	0.9974	0.9949	0.2005	0.9229
$Recall_{negative}$	0.0066	0.0033	0.8026	0.1557
$MacroRecall$	0.3376	0.3389	0.5911	0.4112
$F1_{positive}$	0.0174	0.0366	0.7731	0.2623
$F1_{neutral}$	0.3010	0.3008	0.2644	0.3170
$F1_{negative}$	0.0130	0.0065	0.7251	0.2630
$MacroF1$	0.4035	0.4131	0.5996	0.4972

The results show that the proposed method is good at classification which attaches more importance to precision. The results of 5-classes experiment also showed that the proposed method is very good at dealing with extreme sentiment(*"very postive"* and *"very negative"*). Compared with CoreNLP in 5-classes experiment, the precision of *"very postive"* has been improved by 17.55% and the precision of *"very negative"* has been improved by 15.91%. However, the recall and F1-value of CoreNLP is much more excellent that the prosed method. The recall performance of the proposed method still needs improvement.

6.3.3 Discussion

The recall values of the non-neutral classes of the proposed method are lower than CoreNLP. One of the reasons is, the proposed method depends on the selected sentiment features and sentiment value of the feature words. In the experiment, we choose 3 common features (*"subject"*, *"direct object"* and *"indirect object"*) to construct the model. If the sentiment distribution of a sentence does not cover most of these feature parts, or if the proportion of sentiment distribution of feature words is much less than others, the model will make mistakes. Another reason is, in the experiment, the empty positions of the matrix represented by *"□"* are all set by 0.5, if the selected features can not represent most of the sentiment distribution of the sentence, the measurement result will tend to neutral class. This causes the lower recalls of the non-neutral classes and higher recalls of the

Table 6.2: 5-classes Experiment Results

	CoreNLP	Best 5c Model
$Precision_{verypos]}$	0.6707	**0.8462**
$Precision_{positive}$	0.4095	0.3290
$Precision_{neutral}$	0.3881	0.2078
$Precision_{negative}$	0.4476	**0.4509**
$Precision_{veryneg}$	0.4947	**0.6538**
$MacroPrecision$	0.4821	**0.4975**
$Recall_{verypos}$	0.2807	0.0276
$Recall_{positive}$	0.5901	0.2961
$Recall_{neutral}$	0.2005	0.6992
$Recall_{negative}$	0.7156	0.2686
$Recall_{veryneg}$	0.1685	0.1219
$MacroRecall$	0.3911	0.2827
$F1_{verypos}$	0.3958	0.0534
$F1_{positive}$	0.4835	0.3117
$F1_{neutral}$	0.2644	0.3204
$F1_{negative}$	0.5508	0.3366
$F1_{veryneg}$	0.2513	0.2054
$MacroF1$	0.4319	0.3605

Table 6.3: Best 5c Model on 3-classes Classification Test

	Precision	Recall	F1
positive	**0.7966**	0.4136	0.5445
neutral	0.2078	0.6992	0.3204
negative	**0.7506**	0.3531	0.4802
macro-	0.5850	0.4886	0.5325

neutral classes. And also, the random generated initial values of the NNs also cause the uncertainties of model performance.

Due to these we plan to fix these defects from 3 perspectives. First, we will try more feature designs by adding or changing features. Second, we will try to use some complex method for the value filling of the initial matrix, for example, try to integrate information of both feature elements and non-feature elements. Third, based on current trained models, we will try to determine the relationship between the initial values of the NNs and the trained model performance, and then optimize the initialized setting.

Table 6.4: Performance Improvements on 3-classes test

	Best 3c Model		Best 5c Model	
	SCLM	SCLM(b)	SCLM	SCLM(b)
$Precision_{positive}$	0.1221	0.0399	0.0693	-0.0129
$Precision_{neutral}$	**0.0142**	**0.0142**	0.0306	0.0306
$Precision_{negative}$	**0.2452**	**0.2452**	0.1506	0.1506
$MacroPrecision$	0.1272	0.0998	0.0835	0.0561
$Recall_{positive}$	0.1463	0.1364	0.4048	0.3949
$Recall_{neutral}$	-0.0745	-0.0720	-0.2982	-0.2957
$Recall_{negative}$	0.1491	0.1524	0.3465	0.3498
$MacroRecall$	0.0736	0.0723	0.1510	0.1497
$F1_{positive}$	0.2449	0.2257	0.5271	0.5079
$F1_{neutral}$	**0.0160**	**0.0162**	0.0194	0.0196
$F1_{negative}$	**0.2500**	**0.2565**	0.4672	0.4737
$MacroF1$	0.0937	0.0841	0.1290	0.1194

6.4 Conclusion and Future Work

In this chapter a Sentiment Analysis method was proposed to deal with sentiment measurement and classification using a Modifying-Matrix based Language Model. The regression result shows that the deviation between the output sentiment and target sentiment does not exceed a class distance of five-sentiment-class range. The classification experiment shows that the proposed method has improved most performances than the simplified SCLM, and in some cases it has a higher precision performance. However, the recall performance of the proposed method still needs improvement.

The advantage of the SCLM and the proposed method is that it treats all the words which contain sentiment information as modifying, including negative modifying. So we don't need to parse the complex syntax rules of the negative sentences. However, there is a disadvantage that the proposed method ignores the negation of the same vector feature position. For example, by the setting of the experiment, the difference between *"I don't really like it. "* and *"I really don't like it"* can not be recognized, because the word *"really"* and *"don't"* are at the same modifying position.

Due to these we will focus on improving the recall performance and try to recognize the sentiment difference in finer granularity in the future work. We will focus on the relationship between the initial values of the system and the performance after trained.

And also, we will try to propose a complex operational model for matrices of SCLM to keep the advantage of simplified SCLM.

Chapter 7

Contribution and Recommendation

7.1 Summary of full thesis

In view of the three difficulties of short texts in QA system: semantic, knowledge and emotion , this thesis solves five aspects in total:

1. In the application of the actual Chinese question answering system, due to the characteristics of the Chinese language, there are a lot of uncertain expressions in the question and answer sentence, which can be divided into two categories: the uncertainty of knowledge expression and the uncertainty of semantic expression. Because the existing matching degree calculation method is not suitable for a large number of uncertain application scenarios, in order to solve the problem of knowledge expression uncertainty, this thesis proposes a Chinese question and answer retrieval method based on rough set knowledge discovery, which utilizes the attributes of rough sets. The reduction method and the upper approximation concept find and represent knowledge from the labeled question and answer corpus, and then combine the traditional sentence similarity method to calculate the matching degree between the question and the candidate sentence.

2. In Chinese DBQA, the semantic expression uncertainty of the question and answer sentence is mainly caused by two aspects, one is the uncertainty caused by the Chi-

nese word segmentation process, and the other is the rich expression form caused by the Chinese language feature, which leads to semantic expression uncertainty. Therefore, this thesis proposes a new method for Chinese DBQA. The rule acquisition method based on the discrimination matrix is developed, that is, based on the obtained rules, the QA matching pattern obtained by training the QA can be represented by the attribute words obtained after reduction, and then the words are inversely expressed into the QA mode to represent The match score can then be calculated using the attribute words in the test QA pair.

3. In KBQA, knowledge subject extraction is one of the important step for knowledge base tuple association. The tagging and extraction of knowledge subject is similar to syntactic subject tag and named entity tag, and can be regarded as a sequence tagging problem. In this thesis, from the perspective of the importance of rough set attributes, combined with the existing sequential annotation method used in the named entity extraction and syntactic subject recognition methods, a serialized annotation method for knowledge subject extraction in Chinese KBQA system is presented, to improve the ability of existing methods to deal with the expression of uncertainty in Chinese short sentences.

4. In KBQA system, the analysis of knowledge predicate information in a question will have an impact on the overall matching effect of the knowledge tuple. Since there are uncertainties in the expression of knowledge predicate information in Chinese short questions, these uncertainties lead to the inability of existing methods to achieve better results. From the perspective of rough set theory, this thesis proposes an analysis method of knowledge predicate in question, which reduces the weakly related expression of knowledge predicate in question, so that the expression in the question can be strongly related to the knowledge predicate. Then it makes the question more effectively match the knowledge predicates in the knowledge tuple, thereby improving the overall knowledge predicate analysis ability of the system.

5. To improve the sentiment analysis performance of SCLM, a new sentiment analysis method is proposed in this thesis. In the proposed method, a global modifying matrix of a sentence will be constructed and determinant value of this matrix will be

calculated and adjusted, and then the final value will be used as the sentiment value of the sentence. The regression experiment shows that the deviation between the output sentiment and target sentiment does not exceed a class distance of 5-classes. The classification experiment shows that the proposed method has improved most of the performance comparing to the simplified SCLM.

7.2 Future Directions

There are still some details to be solved in the method proposed in this thesis. For example, the parameter validation problem of the knowledge discovery method mentioned at the end of the chapter 2, resource consumption problems with pattern matching methods mentioned at the end of the chapter 3, and so on. Since the number of different types of questions and answers in the existing corpus is still small, solving these problems still requires enriching and perfecting the existing corpus.

This thesis uses only the relevant corpus of two open domain Chinese question and answer systems. The existing open source QA corpus is mainly related to English, and the number of relevant open source corpora for the open field Chinese question answering system is very rare. Therefore, the application of some methods in other types of corpus and question answering systems has not been developed. For example, the two different processing methods for tags mentioned in chapter 3 have certain performance differences in the corpus used in the experiment, and whether the two tag processing methods will have similar performance differences in other corpora, there are currently no other suitable corpora to test.

Due to research time and research resources limitations, a complete QA system has not yet been implemented. In addition, there is also a lack of certain interaction between the five methods proposed in this thesis. For example, the application of knowledge subject and knowledge predicate to the fast matching of the whole mass knowledge base tuple has not been attempted. Therefore, the implementation of the complete question and answer system will be carried out in the future.

Bibliography

[1] Yorick Wilks, Roberta Catizone, Simon Worgan, and Markku Turunen. Review: Some background on dialogue management and conversational speech for dialogue systems. *Computer Speech & Language*, 25(2):128–139, 2011.

[2] Weizenbaum and Joseph. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[3] C. O. Kohler, G Wagner, and U Wolber. Interactive data processing in medicine–man-machine dialogue. *Methods of Information in Medicine*, 15(02):102–120, 1976.

[4] Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57–66, 1997.

[5] Christiane Fellbaum. Wordnet. *Theory & Applications of Ontology Computer Applications*, pages 231–243, 2010.

[6] Volker Klingspor, John Demiris, and Michael Kaiser. Human-robot-communication and machine learning. *Applied Artificial Intelligence*, 11(7):719–746, 1997.

[7] R. T. Morris and B Samadi. Neural network control of communications systems. *IEEE Transactions on Neural Networks*, 5(4):639–50, 1994.

[8] Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research*, 16(1):105–133, 2011.

[9] Mihai Surdeanu, Mihai Surdeanu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. *Meeting on Association for Computational Linguistics*, pages 33–40, 2002.

[10] David Griol, Lluís F. Hurtado, Encarna Segarra, and Emilio Sanchis. A statistical approach to spoken dialog systems design and evaluation. *Speech Communication*, 50(8-9):666–682, 2008.

[11] Diane J. Litman and Shimei Pan. Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12(2-3):111–137, 2002.

[12] Adrian Horia Dediu, Joana M. Matos, and Carlos Martín-Vide. Natural language processing, moving from rules to data. *Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, pages 24–38, 2017.

[13] R Sowmya and K R Suneetha. Data mining with big data. *International Conference on Intelligent Systems and Control*, pages 246–250, 2017.

[14] Yu Qing Zhang, Xiao Fei Wang, Xue Feng Liu, and Ling Liu. Survey on cloud computing security. *Journal of Software*, 8271(1):302 – 311, 2016.

[15] Y Lecun, Y Bengio, and G Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[16] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David Mcclosky. The stanford corenlp natural language processing toolkit. *Meeting of the Association for Computational Linguistics: System Demonstrations*, 1:14–19, 2014.

[17] Ted Kwartler. The opennlp project. *Text Mining in Practice with R*, pages 237–269, 2017.

[18] Christian Middleton and Ricardo Baeza-Yates. A comparison of open source search engines. *Grid Computing*, pages 10–13, 2008.

[19] Michael Mccandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0.* Manning Publications Co., London, 2010.

[20] Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. Indri: A language-model based search engine for complex queries. *International Conference on Intelligent Analysis*, 2005.

[21] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase:a collaboratively created graph database for structuring human knowledge. *SIGMOD Conference*, pages 1247–1250, 2008.

[22] S Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web, International Semantic Web Conference, Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November*, pages 722–735, 2007.

[23] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *Computer Science*, 2015.

[24] Samira Shaikh, Tomek Strzalkowski, George Aaron Broadwell, Jennifer Stromer-Galley, Sarah Taylor, and Nick Webb. MPC: A multi-party chat corpus for modeling social phenomena in discourse. *International Conference on Language Resources and Evaluation, LREC*, 2010.

[25] Ellen M Voorhees. The trec question answering track. *Natural Language Engineering*, 7(4):361–378, 2001.

[26] Y. Sasaki. Question answering as abduction: A feasibility study at NTCIR QAC1. *Ieice Transactions on Information & Systems*, 86(9):1669–1676, 2003.

[27] Nan Duan. Overview of the nlpcc-iccpol 2016 shared task: Open domain chinese question answering. *International Conference on Computer Processing of Oriental Languages*, pages 942–948, 2016.

[28] SMP2017, 2017.

[29] Richard S. Wallace. The anatomy of A.L.I.C.E. *Book Chapter of Parsing the Turing Test*, pages 181–210, 2008.

[30] Menno Van Zaanen. Multi-lingual question answering using openephyra. *CEUR Workshop Proceedings*, 2008.

[31] Chatterbot, 2017.

[32] Denis Savenkov and Eugene Agichtein. When a knowledge base is not enough:question answering over knowledge bases with external text data. *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 235–244, 2016.

[33] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1870–1879, 2017.

[34] Diego Molla and Jose Luis Vicedo. Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61, 2007.

[35] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. 18(11):613–620, 1975.

[36] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: a probabilistic analysis. *Seventeenth ACM Sigact-Sigmod-Sigart Symposium on Principles of Database Systems*, pages 159–168, 1998.

[37] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Proc of Advances in Neural Information Processing Systems.*, 26:3111–3119, 2013.

[39] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *Computer Science*, 4(2):1178–1188, 2014.

[40] Eric Miller. An introduction to the resource description framework. *Journal of Library Administration*, 34(3-4):245–255, 2001.

[41] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. Knowledge base completion via search-based question answering. *Proceedings of the 23rd International Conference on World Wide Web*, pages 515–526, 2014.

[42] Michel Chein and Marie Laure Mugnier. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer Publishing Company, Incorporated, New York, 2010.

[43] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. *Meeting of the Association for Computational Linguistics*, pages 956–966, 2014.

[44] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over freebase with multi-column convolutional neural networks. *Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 260–269, 2015.

[45] Jimmy Lin and Boris Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. *Twelfth International Conference on Information and Knowledge Management*, pages 116–123, 2003.

[46] Huan Cui, Dong Feng Cai, and Xue Lei Miao. Research on web-based chinese question answering system and answer extraction. *Journal of Chinese Information Processing*, 18(3):24–31, 2004.

[47] Dongfeng Cai, Huan Cui, Xuelei Miao, Chenguang Zhao, and Xiangshi Ren. A web-based chinese automatic question answering system. *International Conference on Computer and Information Technology*, pages 1141–1146, 2004.

[48] Y. Jun Hu, Jiang J. Xin, and Chang H. You. Chinese short-text classification based on topic model with high-frequency feature expansion. *Journal of Multimedia*, 8(4):425–431, 2013.

[49] Cheng Zhang, Xinghua Fan, and Xianlin Chen. *Hot Topic Detection on Chinese Short Text*. Springer Berlin Heidelberg, Berlin, 2011.

[50] Hui He, Bo Chen, Weiran Xu, and Jun Guo. Short text feature extraction and clustering for web topic mining. *Third International Conference on Semantics, Knowledge and Grid*, 9 2007.

[51] Zdzisław Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, Oct 1982.

[52] Didier Dubois and Henri Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems*, 17(2-3):191–209, 1990.

[53] Qinghua Hu, Daren Yu, Jinfu Liu, and Congxin Wu. Neighborhood rough set based heterogeneous feature subset selection. *Information Sciences*, 178(18):3577–3594, 2008.

[54] Wojciech Ziarko. Variable precision rough set model. *Journal of Computer & System Sciences*, 46(1):39–59, 1993.

[55] Z. Zhang, D. Miao, and X. Yue. Similarity measure for short texts using topic models and rough sets. *Journal of Computational Information Systems*, 9(16):6603–6611, 2013.

[56] Nguyen Chi Thanh and Koichi Yamada. Document representation and clustering with wordnet based similarity rough set model. *International Journal of Computer Science Issues*, 8(5):1–8, 2011.

[57] Tuan Fang Fan and Churn Jung Liau. Rough set-based concept mining from social networks. *IEEE International Conference on Fuzzy Systems*, pages 663–670, 2016.

[58] Lixia Cao, Guangqiu Huang, and Weiwen Chai. A knowledge discovery model for third-party payment networks based on rough set theory. *Journal of Intelligent & Fuzzy Systems*, 33(1):413–421, 2017.

[59] Rong Dai and Xiangmin Duan. *Research on Knowledge Acquisition of Motorcycle Intelligent Design System Based on Rough Set*. Springer Berlin Heidelberg, Berlin, 2012.

[60] Xue Gang Chen, Sheng Duan, and Lu-Da Wang. Research on trend prediction and evaluation of network public opinion. *Concurrency & Computation Practice & Experience*, 29(4):4212–4221, 2017.

[61] Yiyu Yao and Yan Zhao. Attribute reduction in decision-theoretic rough set models. *Information Sciences*, 178(17):3356–3373, 2008.

[62] Guangming Lang, Duoqian Miao, Tian Yang, and Mingjie Cai. Knowledge reduction of dynamic covering decision information systems when varying covering cardinalities. *Information Sciences*, 346-347:236–260, 2016.

[63] Andrzej Skowron and Cecylia Rauszer. The discernibility matrices and functions in information systems. *Intelligent Decision Support Handbook of Application & Advance of the Rough Set Theory*, 11:331–362, 1992.

[64] D. Q. Miao, Y. Zhao, Y. Y. Yao, H. X. Li, and F. F. Xu. Relative reducts in consistent and inconsistent decision tables of the pawlak rough set model. *Information Sciences*, 179(24):4140–4150, 2009.

[65] Wen Xiu Zhang, Ling Wei, and Jian Jun Qi. Attribute reduction in concept lattice based on discernibility matrix. *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 157–165, 2005.

[66] Ming Yang. An incremental updating algorithm for attribute reduction based on improved discernibility matrix. *Chinese Journal of Computers*, 30(5):815–822, 2007.

[67] L. U. Xiaohong, Shiquan Chen, and W. U. Jinpei. Heuristic attribute reduction based on discernibility matrix and its application. *Computer Engineering*, 29(1):56–59, 2003.

[68] Ke Wang and Qi Bing Zhu. A heuristic algorithm for attribute reduction based on the discernibility matrix. *Computer Engineering & Science*, 30(6):73–75, 2008.

[69] Ping Yang, Jisheng Li, and Yongxuan Huang. An attribute reduction algorithm by rough set based on binary discernibility matrix. *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 276–280, 2008.

[70] Ming Yang, Songcan Chen, and Xubing Yang. A novel approach of rough set-based attribute reduction using fuzzy discernibility matrix. *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 96–101, 2007.

[71] Fen Shu. Improvement of rule discernibility matrix and method for attributes value reduction. *Computer Engineering & Applications*, 43(32):77–79, 2007.

[72] Baowei Zhang. A new heuristic algorithm for attribute value reduction based on discernibility matrix. *Computer Applications & Software*, 2010.

[73] Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma. Sentiment analysis: A review and comparative analysis of web services. *Information Sciences*, 311:18–38, 2015.

[74] Fuji Ren and Ye Wu. Predicting user-topic opinions in twitter with social and topical context. *IEEE Transactions on Affective Computing*, 4(4):412–424, 2013.

[75] G Vinodhini and RM Chandrasekaran. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6), 2012.

[76] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.

[77] Cynthia Breazeal. *Affective interaction between humans and robots*. Springer, 2001.

[78] S Chandrakala and C Sindhu. Opinion Mining and sentiment classification: a survey. *ICTACT journal on soft computing*, 2012.

[79] Anna Stavrianou, Periklis Andritsos, and Nicolas Nicoloyannis. Overview and semantic issues of text mining. *ACM Sigmod Record*, 36(3):23–34, 2007.

[80] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[81] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[82] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 786–794, 2010.

[83] Zitao Liu, Wenchao Yu, Wei Chen, Shuran Wang, and Fengyi Wu. Short text feature selection for micro-blog mining. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1–4. IEEE, 2010.

[84] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2330–2336. AAAI Press, 2011.

[85] Zhendong Dong and Qiang Dong. HowNet-a hybrid language and knowledge resource. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, pages 820–824. IEEE, 2003.

[86] George A Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[87] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC*, volume 10, pages 1320–1326, 2010.

[88] Changqin Quan and Fuji Ren. Feature-level sentiment analysis by using comparative domain corpora. *Enterprise Information Systems*, pages 1–18, 2014.

[89] Liu Wenyin, Xiaojun Quan, Min Feng, and Bite Qiu. A short text modeling method combining semantic and statistical information. *Information Sciences*, 180(20):4031–4041, 2010.

[90] Duoqian Miao, Feiran Zheng, Zhifei Zhang, and Can Gao. News Topic Detection Approach on Chinese Microblog. *Computer Science*, 1:033, 2012.

[91] Lei Wang, Duoqian Miao, and Cairong Zhao. Chinese Emotion Recognition Based on Three-Way Decisions. In *Rough Sets and Knowledge Technology*, pages 299–308. Springer, 2015.

[92] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 241–249. Association for Computational Linguistics, 2010.

[93] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[94] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. BTM: Topic modeling over short texts. *Knowledge and Data Engineering, IEEE Transactions on*, 26(12):2928–2941, 2014.

[95] Ou Jin, Nathan N Liu, Kai Zhao, Yong Yu, and Qiang Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 775–784. ACM, 2011.

[96] Brendan; Ramnath Balasubramanyan; Bryan R. Routledge O'Connor and Noah A. Smith. From tweets to polls: linking text sentiment to publid opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 122–129. AAAI Press, 2010.

[97] Nicolas Pröllochs, Stefan Feuerriegel, and Dirk Neumann. Negation scope detection in sentiment analysis: Decision support for news-driven trading. *Decision Support Systems*, 2016.

[98] Nicolas Pröllochs, Stefan Feuerriegel, and Dirk Neumann. Detecting negation scopes for financial news sentiment using reinforcement learning. In *Hawaii International Conference on System Sciences*, pages 1164–1173, 2016.

[99] Po Cheng Huang, Jain Shing Wu, and Chung Nan Lee. Negative emotion event detection for chinese posts on facebook. In *International Conference on Cloud Computing and Big Data*, pages 329–335, 2015.

[100] Jian-Yun Nie Xiaodong Yue. Zhifei Zhang, Duoqian Miao. Sentiment uncertainty measure and classification for negative sentences. *Journal of Computer Research and Development*, 2015.

[101] Wen Yi Huang and Tsang Long Pao. A study on the combination of emotion keywords to improve the negative emotion recognition accuracy. In *Information Science and Service Science and Data Mining*, pages 499 – 503, 2012.

[102] Xiaodong Zhang, Houfeng Wang, Li Li, Maoxiang Zhao, and Quanzhong Li. *Negative Emotion Recognition in Spoken Dialogs*. 2015.

[103] Zhao Han, Fuji Ren, and Duoqian Miao. Synthetic and computational language model for interactive dialogue system. In *European Conference on Data Mining 2015 and International Conferences on Intelligent Systems and Agents 2015 and Theory and Practice in Modern Computing 2015. Proceedings*, pages 73–80, New York, NY, USA, 2015.

[104] Gabor Angeli and Christopher D Manning. NaturalLI: Natural Logic inference for common sense reasoning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[105] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

[106] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[107] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.

[108] SB Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. In *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24. IOS Press, 2007.

[109] M Vaggi, S Prandin, G Bellentani, A Costantini, A Labib, et al. Supervised Neural Network Algorithms: A Proved Successful Methodology To Identify Unconventional Layers Opens New Potential Development Scenarios Of Abu Qir Fields In The Mediterranean Sea Offshore Egypt. In *Offshore Mediterranean Conference and Exhibition*. Offshore Mediterranean Conference, 2013.

[110] Judith E Dayhoff. *Neural network architectures: an introduction*. Van Nostrand Reinhold Co., 1990.

[111] James A Anderson. *Neurocomputing*, volume 2. MIT press, 1993.

[112] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[113] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[114] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[115] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989.

[116] Fredric M Ham and Ivica Kostanic. *Principles of neurocomputing for science and engineering*. McGraw-Hill Higher Education, 2000.

[117] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL*, pages 151–161, 2011.

[118] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.

[119] Muneki Yasuda and Kazuyuki Tanaka. Approximate learning algorithm for restricted boltzmann machines. In *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on*, pages 692–697. IEEE, 2008.

[120] JD Paola and RA Schowengerdt. A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *International Journal of remote sensing*, 16(16):3033–3058, 1995.

[121] Zdzislaw Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, 1992.

[122] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. *THE LREC 2010 WORKSHOP ON NEW CHALLENGES FOR NLP FRAMEWORKS*, pages 45–50, 2010.

[123] Zdzisław Pawlakab. Rough set approach to knowledge-based decision support. *European Journal of Operational Research*, 99(1):48–57, 1997.

[124] Sojka P. Rehurek R. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.*, 2010.

[125] Hua Ping Zhang, Hong Kui Yu, De Yi Xiong, and Qun Liu. Hhmm-based chinese lexical analyzer ictclas. *Sighan Workshop on Chinese Language Processing*, pages 184–187, 2003.

[126] Xiangzhou Huang, Baogang Wei, and Yin Zhang. Automatic question-answering based on wikipedia data extraction. *International Conference on Intelligent Systems and Knowledge Engineering*, pages 314–317, 2016.

[127] Nlpcc2016kbqa(tagver), 2017.

[128] Fabian Pedregosa, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and Jake Vanderplas. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(10):2825–2830, 2011.

[129] John HENRY WILKINSON, Friedrich Ludwig Bauer, and C Reinsch. *Linear algebra*, volume 2. Springer, 2013.

[130] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[131] Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing With Compositional Vector Grammars. In *In Proceedings of the ACL conference*. Citeseer, 2013.

[132] Shengping Jin, Fangfang Xiong, and Qiong Li. Conditions and Judgements of Matrices with Positive Real Eigenvalues. *Journal of Chongqing University of Technology(Natural Science)*, 25(1):117–119, 1 2011.

[133] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.

[134] Rajendra Bhatia. *Positive definite matrices*. Princeton University Press, 2009.

[135] Tom Schaul, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, Thomas Rückstieß, and Jürgen Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

[136] Shunji Uchimurai, Yoshihiko Harnamotoi, and Shingo Tomitai. On the Effect of the Nonlinearity of the Sigmoid Function in Artificial Neural Network Classifiers. In *1995 IEEE International Conference on Neural Networks: Proceedings, the University of Western Australia, Perth, Western Australia, 27 November-1 December 1995*, volume 1, page 281. IEEE, 1995.

[137] Bruce Curry and Peter Huw Morgan. Neural networks, linear functions and neglected non-linearity. *Computational Management Science*, 1(1):15–29, 2003.

[138] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.