Question Classification Based on an Extended Class Sequential Rule Model

Zijing Hui, Juan Liu*, Lumei Ouyang
Computer School, Wuhan University, Wuhan 40072, P. R. China
{zijinghui, liujuan, ouyanglumei}@whu.edu.cn

Abstract

Question classification is a crucial preprocessing for question answering system; it can help to make sure the user's intention. Most of previous researches focus on the feature driven methods that represent a question with a bag of features, which ignore the important information contained in the words order and distance. To take such information into account, this paper proposes to describe the questions via the ExCSR (Extended Class Sequential Rule) model. To mine ExCSR rules, a method based on PrefixSpan, called DS-SRM (Distance Sensitive Sequential Rule Miner), is presented as well. Due to the existence of redundancy in the mined rules, a rule selection algorithm MCRSelection (Most Cover Rule Selection) is also proposed to find the most interesting rules. Experiments results on UIUC question set[1] show that the proposed method can achieve the accuracy of 90.6%, which outperforms the previously reported results.

1 Introduction

Question classification, i.e., classifying questions into predetermined types, is quite an important problem in question answering (QA) system, it helps to make clear the intention of users so that the system can choose the appropriate strategies of answers searching and ranking. Similar to other classification problems, question classification usually needs to build the classifier from the training data which contains a set of labeled questions; the classifier is then used to classify the unlabeled questions.

Although questions are special kinds of texts, question classification is more challenging than text classification. Compared to normal texts, a question is usually a very short sentence (some even with few of words), and mostly one word just occurs once in one question, which results that the widely used vector space model (mainly based on TF/IDF) fails to work. To address this problem, researchers usually develop a lot of features, such as location, organization, name, etc., to annotate the words/phrases in the questions, and then use the bag of features to represent the questions. However, such methods still suffer some problems. (1) To get the satisfactory classification performances, they usually need an extremely large amount of features. For example, Li et al. used more than 200,000 features (Li et al., 2006) to represent questions in UIUC question set, while Huang et al. used 13,697 binary features in their best feature space (Huang et al., 2008). (2) The bag of features representation ignores the relationship and the order information among words within the questions, which will cause the misclassification. For example, "Which city is famous for rose?" and "Which rose is famous for city?" belong to two different classes (the former one asks about the <location>, while the later one is about the <entity> or <plant>), due to that "rose" and "city" have different orders. (3) The information of word distance is valuable in question classification yet is not considered by the present methods. For example, "How many people did Randy Craft kill?" asks about the <number>, while "How Randy Craft killed many people?" is about the <description>. The difference of the distances between "how" and "many" plays a role on the types of the questions. CSRs (Class Sequential Rules) originally proposed for opinion extraction (Hu and Liu, 2006) take the word sequences into account. However, it still ignores the distance information between words.

In this paper, we extend the representation of CSRs by integrating the distance information into it, and propose the ExCSR (Extended Class Sequential Rule) model. To mine the ExCSRs, we further propose an algorithm, called DS-SRM (Distance Sensitive Sequential Pattern Miner), based on PrefixScan (Pei et al., 2004). To remove the redundancy of the mined rules, we present an efficient rule selection method, MCRSelection (Most Cover Rule Selection). The remainder of this paper is organized as follows: section 2 introduces the related works; section 3

* the corresponding author

[1] http://cogcomp.cs.illinois.edu/Data/QA/QC/

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 938–946,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

describes the CSR and its extension ExCSR; the rule mining and selection algorithms are presented in section 4; section 5 is the evaluation experiments and results; the paper ends with the concluding remarks in the last section.

2 Related Works

Question classification is a process that assigns a question to a single category or a set of categories. The categories can be organized as either flat or hierarchical taxonomies. Li & Roth (2002) defined a two-layered taxonomy shown in Table 1. The taxonomy consists of six coarse categories and a total of 50 finer categories. Since this taxonomy has been regarded as somewhat informal standard and has been used in much other work on question classification, it is also used in our paper. Given the taxonomy, the classification machinery is then needed to put the questions into specific category or categories. There are two main machineries for the classification, i.e., manual and automatic. The manual methods (Hermjakob, 2001) use hand-written rules and heuristics to do the classification, thus it is time consuming and hard to extend to new question categories; while the automatic methods classify the questions based on machine learning technologies or statistical methods, thus are much more efficient and easy to extend to new question types. Therefore, most of the work follows the automatic methods, which will be also adopted in our work. There are many machine learning methods have been proposed for automatic question classification. Radev et al. (2002) proposed to learn rules by using decision tree method, after trained on TREC-8 and TREC-9, it reached an accuracy of around 70% on TREC-10. Li and Roth (2002) reported a hierarchical approach based on the SNoW learning architecture. By trained on 5500 questions and tested on 500 questions from TREC-10, which are collected in UIUC dataset, it reached an accuracy of 84.2%. Zhang and Lee (2003) used linear SVMs with all possible question word grams, and obtained accuracy of 79.2%. Krishnan et al. (2005) used a short sequence called informer span as important features that are identified by the Conditional Random Field (CRF), and built a meta-classifier using a linear SVM on the features. Their model got the accuracy of 86.2% on UIUC question set. Li and Roth (2006) used more semantic information in WordNet, plus their originally proposed syntactic ones (Li & Roth, 2002), after being trained on 21500 questions, their model achieved an accuracy of 89.3% on a test set of 1000 questions. Li et al. (2008) propose to classify what-type questions by head noun tagging and achieve 85.60% accuracy. Huang et al. (2008; 2009) used much more compact feature set than Li and Roth's work, by taking the head words and their hypernyms as features, with other standard features such as unigrams, they obtained accuracy of 89.2% using linear support vector machine (SVMs), and 89.0% using Maximum Entropy (ME) model. Ray et al. (2010) used the semantic features of the WordNet and the vast knowledge repository in Wikipedia to build the classification model. They trained their model over 5500 questions in UIUC, and tested it over 2393 questions from five TREC collections, and got the average precision of 89.55%. However, to our best knowledge, all of the present works seldom consider the word sequence and word distance for question classification problem. In this paper, we exploit the information of word sequence and word distance in the questions and develop an efficient classification method. The details of the proposed methods will be provided in the next sections.

Coarse Class	**Fine Class**
ABBREVIATION	abbreviation, exp
DESCRIPTION	definition, description, manner, reason
ENTITY	animal, body, color, creative, currency, disease/medicine, event, food, instrument, language, letter, other, plant product, religion, sport, substance, symbol, technique, term, vehicle, word
HUMAN	group, individual, title, description
LOCATION	city, country, mountain, other, state
NUMERIC	code, count, date, money, order, other, period, percent, speed, temperature, volume/size, weight

Table 1. Two-layered taxonomy proposed by Li & Roth (2002)

3 Class Sequential Rule Model and Its Extension

Class sequential rule (CSR) model was originally proposed to represent the labeled sequences in the research of opinion feature extraction (Hu & Liu, 2006). For the completeness of this paper, we first introduce CSR model, then state its extension model ExCSR in this section.

3.1 Class Sequential Rule Model

Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items, and an *element* or *itemset* be a non-empty set of items. A *sequence* is defined as an ordered list of elements, denoted by $\langle e_1 e_2 ... e_r \rangle$, where e_i is an *element* (Liu, 2007). An item can occur only once in an element of a sequence, but can occur multiple times in different elements.

A sequence $s_1 = \langle a_1a_2...a_r\rangle$ is a *subsequence* of another sequence $s_2 = \langle b_1b_2...b_m\rangle$ or s_2 *contains* s_1, if there exist integers $1 \leq j_1 < j_2 < ... < j_{r-1} < j_r \leq m$ such that $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$, ..., $a_r \subseteq b_{jr}$.

Let $D = \{(s_1, y_1), (s_2, y_2), ..., (s_n, y_n)\}$ be the input data of labeled sequences, where s_i is a sequence and $y_i \in Y$ is its class, Y is the set of all classes, $I \cap Y = \emptyset$. A CSR is a production rule $X \rightarrow y$, where X is a sequence, and $y \in Y$.

Table 2 lists some examples of CSRs, where each English word is an item and the class label is in the right side.

Id	Sequence	Class
1	<What difference between>	Desc:desc
2	<Who>	Human:ind
3	<How much weight>	Count:weight
4	<How much>	Count:money
5	<What be NN>	Desc:def

Table 2. Examples of CSRs

A data instance (s_i, y_i) in D is said to cover the CSR $X \rightarrow y$ if s_i contains X. A data instance (s_i, y_i) is said to satisfy *t*he CSR if s_i contains X and $y_i = y$. The *support* of the CSR is the total instances in D that covers the rule. Given the minimum support threshold *min_sup*, a sequence X is called a *sequential pattern* in D if *support(X)* ≥ *min_sup*. The *confidence* of the CSR is the proportion of instances in D that covers the rule also satisfies the rule.

3.2 Extended Class Sequential Rule Model

Although the CSR takes the word sequence into account, it still ignores the distance information between words/phrases which should be also important to question classification. Therefore, we extend the representation of CSRs by considering the distance information. The extended class sequential rule model is called ExCSR hereinafter.

First, we define three simple kinds of distance constraints, shown in Table 3.

Index	Distance Constraint	Description
1	[NEIGH]	Two elements are neighbored. Used to extract the phrase, like “how [NEIGH] much”
2	[NEAR]	Two elements are not more than a give threshold away.
3	[ANY] or blank	Two elements can be of any distance, [ANY] can be omitted.

Table 3. Definition of distance labels

Suppose $<x_1x_2 ... x_r> \rightarrow y$ be a CSR rule, then we define an ExCSR as $<d_1x_1d_2x_2... d_rx_rd_{r+1}> \rightarrow y$, where d_i takes one of the distance constraints in Table 3, which limits the occurring distance between elements x_{i-1} and x_i when the rule is selected to match an instance for classification. For d_1, we can image that there is a special element “x_0” representing the beginning of a sentence, therefore, d_1 is used to constrain the occurring distance of x_1 apart from the beginning of a sentence. Similarly, d_{r+1} constrains the occurring distance of x_r to the end of a sentence.

Then, taking the CSR rules in Table 2 as examples, their extended ExCSR rules might be in the forms listed in Table 4. The support and confidence of an ExCSR is just the same as its original CSR.

Id	Sequence	Class
1	<[NEIGH] What [NEAR] difference [NEAR] between [ANY]>	Desc: desc
2	<[NEIGH] who [ANY]>	Human: ind
3	<[NEIGH] How [NEIGH] much [ANY] weight [NEIGH]>	Count: weight
4	<[NEIGH] How [NEIGH] much [ANY]>	Count: money
5	<[NEIGH] What [NEAR] be [NEAR] NN [NEIGH]>	Desc: def

Table 4. Possible ExCSR examples originated from CSRs in Table 2.

It is noticeable that if only [ANY] is used, then the ExCSR model is actually the same as CSR model. Furthermore, by using ExCSR model, more compact rules that are usually ignored in CSRs mining but play important roles in question classification will be considered. For example, the CSR “<in what year> → date” is too short that it may cover many questions belonging to different classes which causes its confidence may be lower than the threshold; while its corresponding ExCSR “< [NEIGH] in [NEIGH] what [NEIGH] year [ANY] > → date” cannot cover so many questions with conflicted classes as the CSR due to the distance constraint, thus it should have higher confidence and be included in the final classification model.

4 DS-SRM: ExCSRs Mining Method

DS-SRM consists of three parts: preprocessing, mining and rules filtering, shown in Figure 1.

Given a set of question texts, the preprocessing step parses every sentence and translates it into a sequence of elements labeled with se mantic information, and the mining step generates a set of ExCSRs satisfying the minimum support (*min_sup*) and minimum confidence (*min_conf*) requirements from the sequences. Since the distance label [ANY] covers [NEIGH] and [NEAR], the mined ExCSR set should contains a lot of redundant rules with the same element sequences but different distance labels. Therefore, the filtering step is required to remove the redundancy of the rule set. The filtered ExCSRs are to be used to classify the unseen question sentences.

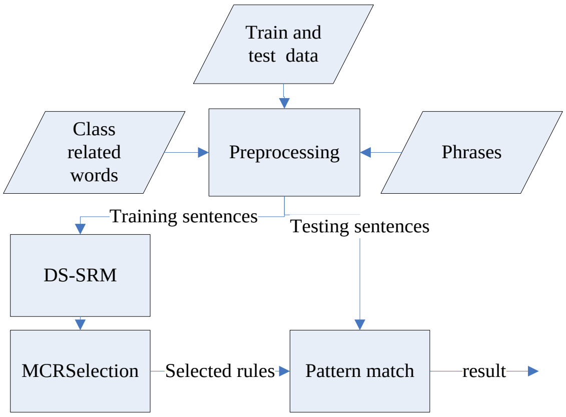

Figure 1. The workflow of DS-SRM

4.1 Preprocessing the Question Texts

Now that the question sentences are in the form of raw texts and cannot be directly used for the rules mining purpose, it's necessary to do the preprocessing by scanning each sentence to do annotation, chunking, and so on. By doing this, each sentence is re-organized by a set of elements with semantic information, especially with the words related to class information. This paper performs the following main preprocessing:

(1) Phrases recognition

There are some phrases consisting of several noun words, which can represent entity classes as a whole, yet each word of it may has different meanings. For example, "*state flower*" denotes a kind of "plant" where "state" is just as a modifier of "flower". While "state" could be regarded as a "location" and "flower" is a kind of "plant" if "state" and "flower" are separately considered. Therefore, we need to recognize such kinds of phrases to avoid ambiguity. We first collect a lot of frequent phrases by grouping the adjacently occurring words from the web pages according to the mutual information statistics. And then locate the ambiguous phrases within the question sentences and manually annotate them with the related class labels (It should be noticed that this processing just recognizes out the phrases of ambiguity, not all phrases or the name entities).

(2) Named entities recognition

A named entity recognizer (NER) of Stanford (Finkel et al., 2005) which defines four types of entities is used to assign a semantic type to noun phrases in a sentence.

For example, a question "*What was W.C. Fields ' real name?*" will be changed into "*What was [person_name] ' real name?*" after the using NER, which will be helpful to assign this question to correct class. However, if we do not tag "*W.C. Fields*" as a person's name, it will be difficult to correctly identify the question's class.

(3) Part-Of-Speech (POS) tagging

POS is important syntactic information in text preprocessing, and it allows us to generate general rules. In our wok, we have used the Stanford Log-linear Part-Of-Speech (Toutanova et al., 2003) to do POS tagging.

(4) Chunking

Besides POS tagging, we also use a chunker, also called as shallow parser, to find out some special structures and phrases and then to eliminate the adjunct words which may have bad impacts on the classification.

For example, in the following question: *What is a group of turkeys called?* (Huang, et al. 2008)

The word "*turkeys*" acts as the central word and contributes to classifying question as "animal", however, the phrase "*a group of*" would introduce ambiguity to misclassify this question to "human group". Therefore, we need to deal with such kinds of phrases. In this work, we adopt the Illinois chunker package (Mavronicolas et al., 1991).

(5) Class related words tagging

For each type of question, there are usually some words related to the question class. Li et al., (2002) have built a list of related words for each question class in their research. For example, for class "*food*", the related words set would be {*alcoholic, apple, beer, berry, breakfast, brew, butter, candy, cereal, champagne, cook, eat, sweat...*}. Using the class label to tag such related words would be very useful for the question classification. However, the class related words may be nouns, verbs and adjectives, and we have found that the verbs and adjectives would cause the ambiguity in our test. Therefore, we only use the nouns in the class related word sets collected by Li et al. (2002). Nevertheless, there still maybe exist the word ambiguity problem. At present, we just ignore it for it has less impact on the performance of our ExCSR rules.

After the preprocessing, the each question sentence will be transformed into a sequence of triplets <[pos], word, [cid]>, where "pos" and "cid" are the POS and class tags of the word respectively. For those *wh* words ("what", "when", "who", "how"…), the "pos" in the triplet is left blank; for any words except those class related nouns, the "cid" is left blank. From the sequences, we can mine out the ExCSRs.

4.2 Mining the ExCSR Rules

The mining procedure is extended from PrefixSpan (Pei et al., 2004), an efficient sequence rule mining method. We will not introduce the PrefixSpan method itself, which is beyond the topic of this paper. Instead, we present our modified version based on the framework of PrefixSpan. The mining algorithm, called DS-SRM (Distance Sensitive Sequential Rule Miner), is shown in Algorithm 1, which is composed of a recursive procedure DS-SPM (Distance Sensitive Sequential Pattern Mining).

It is noticeable that the imported distance information [NEIGH], [NEAR] and [ANY] are not exclusive: [ANY] covers [NEIGH] and [NEAR]; and [NEAR] covers [NEIGH]. Therefore, when counting the occurrence for each item in Step 1 of Procedure DS-SPM, if the current item is (NEIGH, b), then the counters for (NEIGH, b), (NEAR, b), (ANY, b) increase one; if the current item is (NEAR, b), then the counters for (NEAR, b), (ANY, b) increase one.

```
Algorithm 1: (DS-SRM)Distance Sensitive Sequential Rule Miner
Input: The training set $D = \{(s_1, y_1), (s_2, y_2), ..., (s_n, y_n)\}$, where each $s_i$ is the preprocessed sequence;
          The minimum support threshold: min_sup;
          The minimum confidence threshold: min_conf.
          //Rules with the confidence below min_conf or support below min_sup will be disregarded.
Output: A set of ExCSRs satisfying the support and confidence requirements
Parameters: $\Lambda$ -  a sequential pattern set;
                    $\alpha$ - a sequential pattern in $\Lambda$;.
Step 1: $S = \{s_1, s_2, …, s_n\}$; $Y = \{y_1, y_2, …, y_n\}$
Step 2: $\Lambda$ = DS-SPM(<>, 0, S);
Step 3: for each $\alpha$ in $\Lambda$
  (a)  count the frequencies of all covered classes (the classes that the covered sequences belong to),  and find
       the most frequent class label $y \in Y$;
  (b)  if support($\alpha \to$y) $\geq$ min_sup and confidence($\alpha \to$y) $\geq$ min_conf  then output $\alpha \to$y.
```

```
Procedure DS-SPM ($\alpha$, len, $S|_\alpha$)
Input: A sequence set S;
          The minimum support threshold: min_sup;
Output: The complete set of sequential patterns
Parameters: $\alpha$ - a sequential pattern, where each element is a triplet originated from  section 4.1 and have
                  attached the distance information in the form of [(d, pos), (d, word), (d, cid)] (d is the distance
                  information and initially is blank), each pair in a triplet is regarded as an item;
                 len - the length of $\alpha$;
                 $S|_\alpha$ - the $\alpha$-projected sequence set, the collection of labeled suffixes of sequences in S with re-
                  gards to prefix $\alpha$. If $\alpha$ is empty, then $S|_\alpha$= S.
                 Tnear - the threshold to indicate whether two triplets are NEAR.
Step 1: Scan $S|_\alpha$ once to find each frequent item, $(d_b,b)$, such that
a)    $(d_b,b)$ can be assembled to the last element of $\alpha$ to form a sequential pattern; or
b)    $<(d_b,b)>$ can be appended to the last item of $\alpha$ to form a sequential pattern ($<(d_b,b)>$ denotes the triplet
      containing $(d_b,b)$).
Step 2: for each frequent item $(d_b,b)$, append it to $\alpha$ to form a sequential pattern $\alpha$', and output $\alpha$'.
Step 3: for each $\alpha$', construct $\alpha$'-projected set $S|_{\alpha'}$  by the following ways:
(a)  $S|_{\alpha'}$ = set of suffixes of sequences in S with regards to prefix $\alpha$';
(b)  For each item  $(d_c,c)$ in $S|_{\alpha'}$, revising its distance information by one of the following ways:
     (i)   if  $(d_c,c)$ is in the same triplet with $(d_b,b)$, then modify $(d_c,c)$ to $(b,c)$;
     (ii)  if $(d_c,c)$'s triplet is neighbor to  $(d_b,b)$'s triplet, then modify $(d_c,c)$ to (NEIGH, c);
     (iii) if $(d_c,c)$'s triplet is near to  $(d_b,b)$'s triplet with regards of Tnear, then modify $(d_c,c)$ to (NEAR, c);
     (iv)  OTHERWISE, modify  $(d_c,c)$ to (ANY, c);
Step 4: call DS-SPM($\alpha$',len+1, $S|_{\alpha'}$).
```

Algorithm 1. The flow of DS-SRM algorithm

4.3 Filtering out Redundant ExCSR Rules

Since [NEIGH], [NEAR], and [ANY] are not exclusive, Algorithm 1 may generate quite a lot of redundant rules. In order to remove the redundancy, we first define the interestingness of a rule *r* as Equation (1):

$$\text{Interestingness}(r)=\text{support}(r)*\text{confidence}(r) \quad (1)$$

Then we can rank the rules according to their interestingness. Rules with high interestingness score tend to have high support as well as high confidence, thus should be remained.

Due to the use of overlapping constraints, there are usually some rules with the same support and confidence, exemplified in Table 5.

Rule Id	Class	Conf.	Sup.	Rule
1	DESC :desc	100%	32	<[what][be][difference][between]>
2	DESC :desc	100%	32	<[what][be][difference][NEAR][between]>

Table 5. Example rules with same support and confidence

Rule 1 and 2 are with the same confidence 100% and support 32 thus have the same interestingness. However, rule 1 is more general than rule 2 for it uses less distance constraints, so we prefer to remain rule 1 and discard rule 2. For this purpose, we further to define a measure as Equation (2):

$$\text{Distance\_Constraint}(r) = \sum_{i=1}^{k} \text{label\_value}(i) \quad (2)$$

Where *label_value(i)* denotes the value of the *i-th* distance label (the value of [NEIGH], [NEAR], [ANY] is set to 2, 1, 0.5 separately), and *k* is the total number of distance labels used in rule *r*.

Obviously, shorter rules with simpler distance constraint are more general and preferred.

Therefore, we propose a rule selection algorithm MCRSelection illustrated in Algorithm 2, which will be used iteratively for all classes, one run for one class, to remove the redundant rules.

5 Experiments and Discussions

In order to evaluate our proposed method, we have compared our method to other art-of-state methods on the UIUC question set. We first describe the data sets used in our experiments. In order to investigate which combination of the distance information is optimal to our method, we then test our method with different distance combination. Finally, we compare our method with the optimal distance combination to other representative methods. In the experiments, we set the threshold values of parameters *min_sup*, *min_conf* and *Tnear* as 3, 0.75 and 2 respectively by experience.

```
Algorithm 2: MCRSelection
Input: rule set R = {$r_1$, $r_2$...$r_n$} for specific class;
       The training question set D.
Output: rule set R' without redundancy

Step 1: R' = $\emptyset$;
Step 2: Calculate interestingness for each rule in R;
Step 3: Rank rules according to the interestingness
        and find the rule r with the highest interesting-
        ness. If there are one more than such rules, then
        choose the one with the least Distance-Constraint
        value;
Step 4: R' = R' $\cup$ {r}; R = R – {r};
Step 5: D = D- {instances satisfied by r};
Step 6: if D is empty, then return R';
Step 7: For each r$\in$R, update support(r) with regards
        to the updated D;
Step 8: go to step 2.
```

Algorithm 2. The flow of MCRSelection algorithm

5.1 Data Set

Class	Question Num.	Class	Question Num.
ABBR	9	desc	7
abb	1	manner	2
exp	8	reason	6
ENTITY	94	**HUM**	65
animal	16	group	6
body	2	ind	55
color	10	title	1
creative	0	desc	3
currency	6	**LOC**	81
dis.med.	2	city	18
event	2	country	3
food	4	mount	3
instrument	1	other	50
language	2	state	7
letter	0	**NUM**	113
other	12	code	0
plant	5	count	9
product	4	date	47
religion	0	distance	16
sport	1	money	3
substance	15	order	0
symbol	0	other	12
technique	1	period	8
term	7	percent	3
vehicle	4	speed	6
word	0	temp	5
DESC	138	size	0
def	123	weight	4

Table 6. The composition of TREC 10 test set

In order to facilitate the comparison, similar to previously reported results, we also use the same benchmark UIUC question training and test sets in our experiments, where the questions are labeled as six coarse categories and a total of 50 finer categories. Concretely, we train our model with training set containing 5500 labeled questions and test it on the TREC 10 question set with 500 questions. The composition of the test set is listed in Table 6.

5.2 Investigation on the Distance Combinations

In section 3.2, we have introduced three kinds of distance: [NEIGH], [NEAR] and [ANY]. In order to know whether all of them are necessary in our method, we have tested our method with different distance combination on the same data sets. Table 7 shows the overall performances on 6 coarse classes and 50 fine classes with different combinations separately; and Table 8 presents more details on the six coarse categories.

	ANY	**NEAR+ ANY**	**NEIGH+ ANY**	**All**
6 classes	90.6%	92.8%	92.1%	93.6%
50 classes	83.8%	87.8%	86.8%	90.6%

Table 7. Performances of our method with different distance constration combinations

	ABBR	**ENTITY**	**DESC**	**HUMAN**	**LOC**	**NUM**
ANY	81.8%	73%	82%	95%	86%	92.6%
NEAR+ ANY	90%	76%	87%	97%	87%	93%
NEIGH+ ANY	90%	75%	84%	95%	89%	93%
All	90%	79%	95%	98%	90%	95%

Table 8. More detailed investigation on the distance combinations

Obviously, with all of the three kinds of distance information, our method reaches the best performance. In fact, if only [ANY] is considered, our ExCSR model is just the CSR model without the distance constraints, with which the overall accuracies are 90.6% in coarse classes, and 83.3% in fine grained classes respectively, which are lower than the cases that the distance constraints are considered.

The more detailed results in Table 8 also show that including all of the distance information can get the highest prediction accuracies on all of the six categories. Especially for the DESC class, the previously feature bag based methods usually perform not very good due to the fact that question classes are distance sensitive, while our method with all distance information included can get 95% overall accuracy. For example, "What foods contain vitamin B12?" is labeled as "ENTY: food", while "What is fiber in food?" belongs to "DESC: define" in UIUC data sets. The main difference between above question texts is the distance between two words "what" and "food" that are critical to the classification. Due to ignoring the distance information, the feature-bag based method usually cannot correctly classify the second question and may label it as "ENTY: food". While our method can properly distinguish these two questions for they correspond to different ExCSRs. The similar cases occur in questions of class "DESC: desc".

All in all, our proposed ExCSR including all of the three distance constraints is an effective model to the question classification. Thus in the comparison experiment, we consider all of the distance constraints.

5.3 Comparing with other Methods

By considering all of the distance constraints, the performances on different categories of our method are listed Table 9.

Class	Pre.	Rec.	F	Class	Pre.	Rec.	F
ABBR	90	100	94.7	desc	100	85.7	92.3
abb	100	100	100	manner	100	100	100
exp	100	100	94.1	reason	85.7	83.3	83.3
ENTITY	78.8	87.2	82.8	**HUM**	98.3	87.7	92.7
animal	93.3	87.5	90.3	group	71.4	66.7	72.7
body	100	100	100	ind	94.8	90.9	95.2
color	100	100	100	title	/	/	/
creative	/	/	/	desc	100	100	100
currency	100	83.8	90.9	**LOC**	90.2	91.4	90.8
dis.med.	66.7	100	80	city	100	100	100
event	66.7	100	80	country	100	100	100
food	100	75	85.7	mount	100	100	100
instrument	100	100	100	other	83.9	50	85.1
language	100	100	100	state	85.7	100	100
letter	/	/	/	**NUM**	94.5	92	93.2
other	41.7	50	48	code	/	/	/
plant	83.3	100	90.9	count	81.8	100	100
product	75	75	75	date	95.9	100	100
religion	/	/	/	distance	100	93.8	93.8
sport	100	100	100	money	100	33.3	40
substance	73.7	93.3	82.3	order	/	/	/
symbol	/	/	/	other	85.7	66.7	72.7
technique	100	100	100	period	72.7	75	80
term	58.3	100	73.7	percent	75	100	85.7
vehicle	100	100	100	speed	100	100	100
word	/	/	/	temp	100	100	100
DESC	94.8	92	93.4	size	/	/	/
def	95	92.7	93.8	weight	100	100	100

Table 9. The performances on different categories of our method

In order to know what is the rank of our method, we compare our methods with other representative methods: Zhang and Lee (2003); Huang et al., (2008); Krishnan et al., (2005). Since

Huang et al. (2008) have compared their method with Li et al. (2006) and show that their method is superior, we don’t consider Li et al., (2006) method in this work.

The comparison results are shown in Table 10, showing that our method achieves the best accuracy. Although the results of Huang’s method are competitive to our method, however, they use some manual regular expression patterns. Moreover, Huang et al. used a large amount of features 13697 to construct their model, while our ExCSRs model are more compact and final classifier only has 412 rules with length less than 4.

Method	**6 classes**	**50 classes**
Zhang & Lee, Linear SVM	87.4%	79.2%
Krishnan et al., SVM+ CRF	93.4%	86.2%
Huang et al., Maximum Entropy Model	93.6%	89%
Our method	93.6%	90.6%

Table 10. Overall comparison results with other three methods

Now that the results of Huang’s method are competitive to ours, we further compare it with ours on each fine grained class, and the accuracies are shown in Table 11.

Class	**Huang Method**	**Our Method**	**Class**	**Huang Method**	**Our Method**
ABBR			desc	75	100
abb	100	100	manner	100	100
exp	88.9	100	reason	85.7	85.7
ENTITY			**HUM**		
animal	94.1	93.3	group	71.4	71.4
body	100	100	ind	94.8	94.8
color	100	100	title	/	/
creative	/	/	desc	100	100
currency	100	100	**LOC**		
dis.med.	40	66.7	city	100	100
event	100	66.7	country	100	100
food	100	100	mount	100	100
instrument	100	100	other	83.9	83.9
language	100	100	state	85.7	85.7
letter	/	/	**NUM**		
other	45.5	41.7	code	/	/
plant	100	83.3	count	81.8	81.8
product	100	75	date	95.9	95.9
religion	/	/	distance	100	100
sport	100	100	money	100	100
substance	88.9	73.7	order	/	/
symbol	/	/	other	85.7	85.7
technique	100	100	period	72.7	72.7
term	100	58.3	percent	75	75
vehicle	100	100	speed	100	100
word	/	/	temp	100	100
DESC			size	/	/
def	89	95	weight	100	100

Table 11. Precisions for fine grained question categories with Huang’s method and our method

Table 11 shows that our method can achieve better results on most classes, especially for DESC coarse class that is considered to be difficult to identify (Li et al., 2006). By investigating the questions in that class we found that the question classes are sensitive to the word order and distance. Huang represents a sentence as a bag of features and ignore the relative order and of words their distances, thus performed not very well.

Of course, both our method and Huang’s method show bad performance on Entity: other class, which is also shown to be difficult to identify, for the question texts in “other” class is quite fuzzy, we will put emphasis on this kind of class.

We also analyzed the incorrectly identified questions, and found that there are inherently ambiguity in training and testing questions (see examples in Table 12), which also conforms to Huang et al (2008)’s analysis.

Class	**Rule**
ENTITY:animal	What is a group of frogs called ?
ENTITY:termeq	What are the spots on dominoes called ?
ENTITY:termeq	What 's the term for a young fox ?
ENTITY:animal	What is the scientific name for elephant ?

Table 12. Ambiguous questions in testing set

6 Conclusion

In this paper, we first present ExCSR model for question classification, which is extended from the CSR model by integrating the distance information. Compared to CSR model, ExCSR is more compact intuitive, yet effective; then we describe the ExCSR mining algorithm, DS-SRM, and the rule filtering algorithm MCRSelection. By MCRSelection algorithm, we can keep the most interesting rules with less redundancy. Experiment results on the UIUC question set show that our method outperforms previously reported results.

In the future, we will consider more sophisticated method to address the questions with fuzzy information such as those of “other” class in UIUC data set.

Acknowledgements

The work was partially supported by the National Natural Science Foundation of China (60970063, 60773010), the Ph.D. Programs Foundation of Ministry of Education of China

(20090141110026), and the Fundamental Research Funds for the Central Universities (6081007).

References

Xin Li and D. Roth. 2006. Learning Question Classifiers: the Role of Semantic Information. *Natural Language Engineering*, 12(3):229–249.

Zhiheng Huang, M. Thint, and Z. Qin. 2008. Question classification using head words and their hypernyms. *In Proc. of the EMNLP.* pp.927-936

Minqing Hu and Bing Liu. 2006. Opinion Feature Extraction Using Class Sequential Rules. *In Proc. of AAAI-06.*

Pei, J. Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. 2004. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach*.* *IEEE Transactions on Knowledge and Data Engineering,* 16(10):1-17.

Li, X. and D. Roth. 2002. Learning Question Classifiers. *In Proc. of the 19th international conference on Computational linguistics (COLING '02),* vol. 1: 556–562.

Hermjakob, U. 2001. Parsing and question classification for question answering. *In Proc. of ACL-2001 Workshop on Open-Domain Question Answering.*

Radev, D., Fan, W., Qi, H.,Wu, H. & Grewal, A. 2002. Probabilistic question answering on the web. *In Proc. of the 11th international conference on World Wide Web (WWW2002), Hawaii.*

Zhang D. and W. S. Lee. 2003. Question Classification using Support Vector Machines. *In Proc. of the ACM SIGIR conference on information retrieva l(SIGIR'03)*: 26–32.

V. Krishnan, S. Das, and S. Chakrabarti. 2005. Enhanced answer type inference from questions using sequential models. *In Proc. of the HLT/EMNLP'2005.*

Fangtao Li, Xian Zhang, Jinhui Yuan and Xiaoyan Zhu. 2008. Classifying What-Type Questions by Head Noun Tagging. *In Proc. of the 22nd International Conference on Computational Linguistics (COLING 2008),* pp. 481-488

Zhiheng Huang, Marcus Thint, Asli Celikyilmaz. 2009. Investigation of Question Classifier in Question Answering. *In Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing.* pp. 543–550.

Santosh Kumar Ray, Shailendra Singh, B.P. Joshi. 2010. A semantic approach for question classification using WordNet and Wikipedia. *Pattern Recognition Letters*: 1935-1943.

Bing Liu. 2007. Web data mining: Exploring hyperlinks, contents, and usage data. *Springer-Verlag,* Berlin, Heiderlberg.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *In Proc. of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005).* pp.363-370.

Kristina Toutanova, Dan Klein. 2003. Christopher Manning, and Yoram Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *In Proc. of HLT-NAACL.* pp.252-259.

M. Mavronicolas and D. Roth. 1991. Sequential Consistency and Linearizability: Read/Write Objects. *In Proc. of the 29th Annual Allerton Conference on Communication, Control and Computing.* pp.683-692.