Question Classification using Multiple Classifiers

LI Xin
Computer Science
Engineering Dep.
FUDAN Univ., Shanghai
lixin@fudan.edu.cn

HUANG Xuan-Jing
Computer Science
Engineering Dep.
FUDAN Univ., Shanghai
xjhuang@fudan.edu.cn

WU Li-de
Computer Science
Engineering Dep.
FUDAN Univ., Shanghai
ldwu@fudan.edu.cn

Abstract

The Open-domain Question Answering system (QA) has been attached great attention for its capacity of providing compact and precise results for users. The question classification is an essential part in the system, affecting the accuracy of it. The paper studies question classification through machine learning approaches, namely, different classifiers and multiple classifier combination method. By using compositive statistic and rule classifiers, and by introducing dependency structure from Minipar and linguistic knowledge from Wordnet into question representation, the research shows high accuracy in question classification.

1 Introduction

With the rapid development of the Internet, the capacity of textual information has been greatly improved. How to acquire accurate and effective information has become one of the great concerns among Internet users. Open-Domain Question Answering System (QA) has gain great popularities among scholars who care the above problem (Li, et al. 2002; Moldovan, et al. 2003; Zhang, et al. 2003), for QA can meet users' demand by offering compact and accurate answers, rather than text with corresponding answers, to the questions presented in natural language. Therefore, it saves users' great trouble to find out specific facts or figures from large quantity of texts.

The study of Question Classification (QC), as a new field, corresponds with the research of QA. QC is an essential part of QA, for to correctly answer users' questions, the system has to know what the users are looking for, and it is QC that presents important searching clues for the system. QC can be defined to match a question to one or several classes in K category so as to determine the answer type. Every class presents some semantic restrictions on the answer searching, which serves QA with various strategies in locating the correct answer.

The result of QC can also serve QA in the answer selecting and extract, which influence the performance of QA directly. The first reason is that QC minish searching space. For example, if the system knows that the answer type to the question "*Who was the first astronaut to walk in space?"* is a person's name, it can confine the answer in the names, rather than every word in the texts. The second reason is that QC can determine the searching strategies and knowledge base QA may need. For instance, the question "*What county is California in?*" needs the name of a country as its answer, so system needs the knowledge of countries' name and name entities tagging to identify and testify the place name, while the question "*What is Teflon?*" expects an answer in a sentence or a fragment, in the form of *Teflon is* <.... >. In fact, almost all the QA have the QC module and QC is the one of the most important factors what determines the QA system performance (Moldovan, et al. 2003).

At present the studies on QC are mainly based on the text classification. Though QC is similar to TC in some aspects, they are clearly distinct in that : Question is usually shorter, and contains less lexicon-based information than text, which brings great trouble to QC. Therefore to obtain higher classifying accuracy, QC has to make further analysis of sentences, namely QC has to extend interrogative sentence with syntactic and semantic knowledge, replacing or extending the vocabulary of the question with the semantic meaning of every words.

In QC, many systems apply machine-learning approaches (Hovy, et al. 2002; Ittycheriah, et al. 2000; Zhang, et al. 2003). The classification is made according to the lexical, syntactic features and parts of speech. Machine learning approach is of great adaptability, and 90.0% of classifying accuracy is obtained with SVM method and tree Kernel as features. However, there is still the problem that the classifying result is affected by the accuracy of syntactic analyzer, which need manually to determine the weights of different classifying features.

Some other systems adopting manual-rule method make QC, though may have high classifying accuracy, lack of adaptability, because regulation determination involves manual interference to solve the conflicts between regulations and to form orderly arranged rule base.

The paper combines statistic and rule classifiers, specifically statistics preceding regulation, to classify questions. With rule classifier as supplementary to statistic, the advantages of respective classifier can be given full play to, and therefore the overall performance of the classifier combination will be better than the single one. Moreover, as far as the QC task is concerned, the paper compares various classifier combinations, statistic-rule classifier, voting、Adaboost and ANN. To represent questions, the paper uses dependency structure from Minipar (Lin 1998) and linguistic knowledge from Wordnet (Miller 1995; Miller, et al. 2003). In the following parts of the paper, classifying method and features is first introduced, and then comparisons are made between different type features and between feature combination methods. The comparisons are testified in experiments. The last part of the paper is about the conclusion of the present research and about the introduction of the further work to be done on this issue.

2 Classifying Features

In machine learning method, every question should at first be transformed into a feature vector. Bag-of-word is one typical way of transforming questions, where every feature is one word in a corpus, whose value can be Boolean, showing whether the word is present in questions, and which can also be an integer or a real number, showing the presence frequency of the word. In this paper, every question is represented as a Boolean vector.

1. Bag-of-word: all lexical items in questions are taken as classifying features, because stop-word such as “what” and “is” playing a critical role in QC.

2. Wordnet Synsets: Wordnet was conceived as a machine-readable dictionary. In Wordnet, word form is represented by word spellings, and the sense is expressed by Synsets, and every synset stands for a concept. Wordnet shows both lexical and semantic relationships. The former exists between word forms, while the latter exists between concepts. Among various semantic relations in Wordnet, we choose hypernyms between nouns as our only concern. The classifying features are the senses of the nouns in the sentences and synsets of their hypernyms.

3. N-gram: the model is founded on a hypothesis that the presence of a word is only relevant to the *n* words before it. The frequently used are Bi-gram and Tri-gram, and Bi-gram is chosen as the classifying features in the present research. Compared with word, Bi-gram model investigates two historical records, and reflects the partial law of language. It embodies the features of word order, and therefore it can reflect the theme of the sentence more strongly.

4. Dependency Structure: Minipar is a syntactic analyzer, which can analyze the dependency relation of words in sentences. It describes the syntactic relationships between words in sentences. Such relation is direction-oriented, semantically rather than spatially, namely one word governs, or is governed by, another concerning their syntactic relation. In one sentence (W1W2…Wn), compared with Bi-gram, Dependency structure concerns

WiWj , but not need limitation j= i+1. Obviously, Dependency Relation goes further than Bi-gram in language understanding. Dependency structure is specified by a list of labeled tuples. The format of a labeled tuple is as follows:

label (word pos root governor rel exinfo …)

"Label" is a label assigned to the tuple. If the tuple represents a word in the sentence, label should be the index of the word in the sentence. "Word" is a word in the input sentence. "Pos" is the part of speech. "Root" is the root form. "Governor" if the label of the governor of word (if it has one), "rel" is type of dependency relationship, and "exinfo" for extra information. Minipar output is represented by the word dependency relationship via "governor". Though only 79% of recall and some word relations fail to be analyzed, the accuracy reaches 89%, which guarantees that a large proportion of dependency relations from the output are correct. And the experiment proves that Dependency structure has more classify precision than Bi-gram as classifying feature.

For example, as to the question "*Which company created the Internet browser Mosaic?"* Minipar may produce the following results:

*E0 (() fin C *)*
1 (Which ~ Det 2 det (gov company))
2 (company ~ N E0 whn (gov fin))
3 (created create V E0 i (gov fin))
E2 (() company N 3 subj (gov create) (antecedent 2))
……

According to the tuple, we can get dependency relationships between words in sentences. tuple 1 *(Which ~ Det 2 det gov company)* shows us the det relationship between "which" and "company" in the sentence. Therefore, we can get a words-pair（which company）, and likewise other five pairs of words can be obtained－*(company create)*、*（the Mosaic）*、*(Internet Mosaic)*、*(browser Mosaic)*、*(create Mosaic)*, which will be the item of vector represented the question.

3 Classifying Method Description

3.1 Support Vector Machine (SVM)

SVM is a kind of machine learning approach based on statistic learning theory. SVM are linear functions of the form f (x) = <w•x> +b, where <w•x> is the inner product between the weight vector w and the input vector x. The SVM can be used as a classifier by setting the class to 1 if f(x) > 0 and to -1 otherwise. The main idea of SVM is to select a hyperplane that separates the positive and negative examples while maximizing the minimum margin, where the margin for example x_i is y_i f(x) and $y_i \in$ [-1,1] is the target output. This corresponds to minimizing <w•w> subject to y_i (<w•x> +b) $\geq$ for all *i*. Large margin classifiers are known to have good generalization properties. An adaptation of the LIBSVM implementation (Chang, et al. 2001) is used in the following. Four type of kernel function linear, polynomial, radial basis function, and sigmoid are provided by LIBSVM .

3.2 SVM-TBL QC Algorithm

TBL has been a part of NLP since Eric Brill's breakthrough paper in 1995(Brill 1995), which has been as effective as any other approach on the Part-of-Speech Tagging problem. TBL is a true machine learning technique. Given a tagged training corpus, it produces a sequence of rules that serves as a model of the training data. Then, to derive the appropriate tags, each rule may be applied, in order, to each instance in an untagged corpus.

TBL generates all of the potential rules that would make at least one tag in the training corpus correct. For each potential rule, its improvement score is defined to be the number of correct tags in the training corpus after applying the rule minus the number of correct tags in the training corpus before applying the rule. The potential rule with the highest improvement score is output as the next rule in the final model and applied to the entire training corpus. This process repeats (using the updated tags on the training corpus), producing one rule for each pass through the training corpus until no rule can be found with an improvement score that surpasses some predefined threshold. In

practice, threshold values of 1 or 2 appear to be effective.

Therefore, we present compositive QC approach with rule and statistic learning. At first, questions are represented by Bag-of-word, Wordnet Synsets, Bi-gram, and Dependency structure, and are classified by the same samples and same SVM. Then output of SVM is transformed to the input of TBL, and thus every sample in TBL training data is featured by four-dimensioned vectors, from which a new is obtained as training data of TBL. When the errors produced in initial marking process are corrected in TBL to the greatest extent, a final-classifier is produced as follows (Figure1).

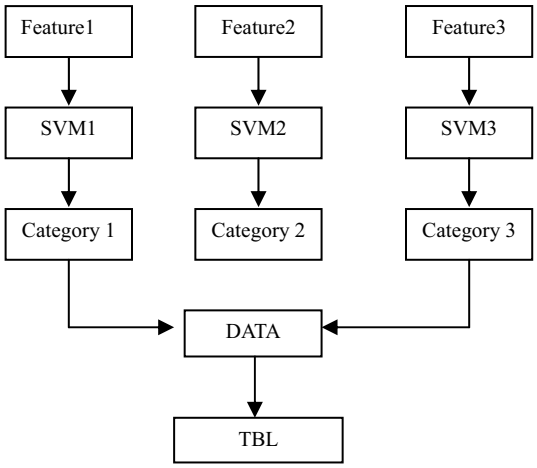

Figure1 SVM-TBL QC Algorithm

TBL is composed of three parts: unannotated text, transformation templates, and objective function. In the experiment, unannotated text is obtained from SVM. The transformation templates define the space of transformations; here is combination of SVM output. Suppose we have k basic classifiers, and each classifier may put questions into N types, then we have $C_k^1 N^1 + C_k^2 N^2 + \cdots + C_k^k N^k$ rule templates. Objective function is the precision of classifier.

4 Results and Analysis

The research adopts the same UIUC data and classifying system as (Zhang, et al. 2003) shows. There are about 5,500 labeled questions randomly divided into 5 training sets of sizes 1,000, 2,000, 3,000, 4,000 and 5,500 respectively. The testing set contains 500 questions from the TREC10 QA track. Only coarse category is test.

4.1 SVM Classifying Result

We experiment the QC by SVM with four kernel function, and the following table (Table1) is the illustration of classifying accuracy by using single-kind classifying feature.

It is shown that as to the four type features, no matter what Kernel is used, using Dependency relation feature have more precision than others and feature of Synsets is better than Bag-of-word. Therefore it is safe to draw the conclusion that Synsets and dependency relationship are helpful to represent questions. Among the four Kernel function, Liner has the best classifying precision. That is why we use Liner in the following experiment.

Num. of Training / Kernel & feature		1000	2000	3000	4000	5500
Liner	Bag-of-word	79.6	81.2	83.4	85.8	84.8
	Wordnet	77.8	83.8	85.2	86.4	86.8
	Bi-gram	73.6	80.6	83.2	87.4	88.6
	Dependency	82.0	86.8	87.2	88.4	89.2
polynomial	Bag-of-word	52.4	69.2	66.0	61.4	62.6
	Wordnet	48.4	69.8	70.0	68.8	73.2
	Bi-gram	27.6	49.2	46.4	49.6	50.8
	Dependency	73.0	78.8	81.8	82.4	85.2
RBF	Bag-of-word	68.8	73.2	80.2	81.4	83.6
	Wordnet	69.0	73.2	79.8	80.2	81.0
	Bi-gram	62.2	70.2	76.0	80.0	81.2
	Dependency	72.8	78.8	81.0	83.2	85.0
Sigmoid	Bag-of-word	65.6	74.2	77.0	78.2	80.2
	Wordnet	74.2	82.6	83.4	83.8	84.4
	Bi-gram	68.6	74.4	79.8	83.2	84.8
	Dependency	75.2	78.0	82.4	83.4	85.2

Table1. Four kernel function Question Classifying Accuracy (%)

4.2 Result of SVM multi-kind-feature classification

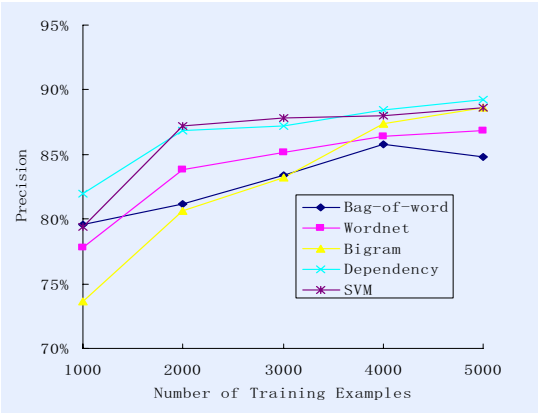

Figure 2. Multi-type Feature

A question can be represented directly as a vector with multi-kind-features: Bag of Word, Dependency Structure, Synonym and Bi-gram. Figure2 provides an accuracy comparison of the results derived from classification with four features and classification with only one kind feature. Experimental result indicates that, results from classification with four type features do not excel the best classification precision with only one feature.

4.3 Using Adaboost to combine several classification results

Multi-classifier combination is often used to obtain better classification results. Adaboost (Schapire 1997; Schapire 1999) is an effective classifier combination method. Yet in question classification training, chances of samples to be faultily classified are slim. Therefore, greater accuracy on classification can hardly be realized with Boost.

4.4 Using BP to combine several classifiers

We have also tried to use nerve network to combine the output results of 4 classifiers. We build a BP network with 4 input nodes and 1 output node. The number of hidden nodes chosen comes from the empirical formula: *m=sqrt(nl)*, whose *"m"* indicates hidden nodes, *"n"* input nodes, and "l" output nodes. Thus, the number of hidden layer nodes is *"2"*.Figure3 shows, when training samples are relatively less, classification accuracy of BP is greater compared to that of single-feature classifier, but not in cases where the number of samples increases.

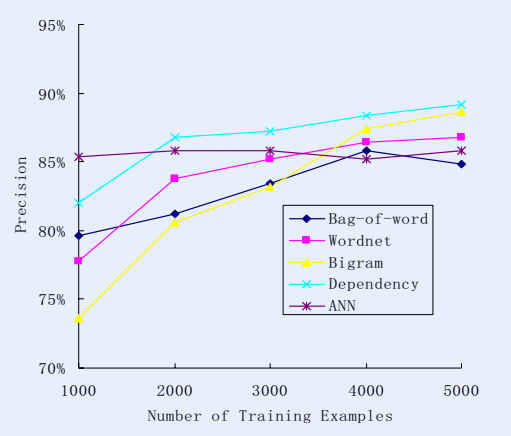

Figure 3. ANN combine several classifiers

4.5 Using the method of voting to combine several classifiers

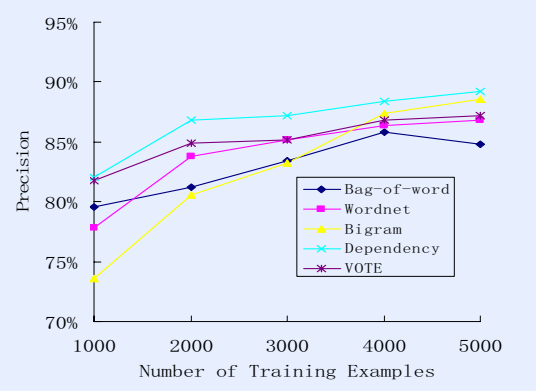

Figure 4. Voting combine several classifiers

Through the method of voting, we can also get the combination results, according to the class label outputted by SVM with different type features. Experimental results are given in Figure 4. We may see that, due to the rule of "more votes winning" in voting, when there are a number of not-so-accurate classifiers, the accuracy of voting can not compete with the greatest accuracy of a single classifier.

4.6 Using TBL method to combine several classification results

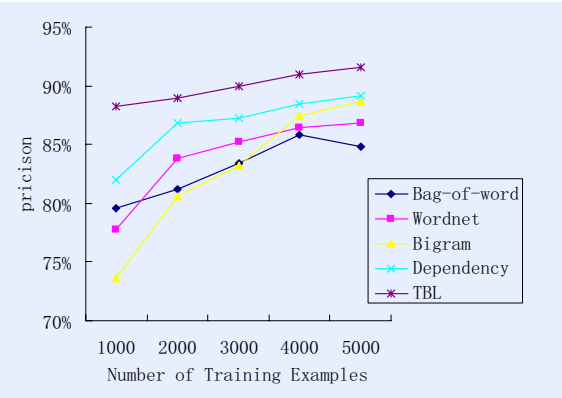

Figure 5. TBL combine several classifiers

Figure 5 displays the accuracy of a number of classification results in TBL combination. In our experiment, we construct 5 test-training sets, using 5500 sentences in UIUC. Each test-training set has 1000 stochastically chosen questions as its test set, and the other 4500 as its training. The TBL training set is built upon the SVM classification results from the test set. In comparison with the method to voting, TBL uses the conversion rule to fully rectify the errors of

initial tagger. Therefore, TBL classification will not produce results inferior to the best results of initial tagging.

We obtain from the experiment all together 251 conversion rules, the foremost ones of which are listed as follows. From these rules which come from TBL training, we may also deduce that, TBL makes use of, firstly, the results of the most accurate classifier (parser), and secondly, the results of other classifiers, especially those of dependency structure rectified by Bi-gram results. It puts the accuracy of SVM single-feature classification into full use to secure greater accuracy.

1. Parser_2 $$ _#=_2
2. Parser_3 $$ _#=_3
3. Parser_1 $$ _#=_1
4. Parser_5 $$ _#=_5
5. Parser_4 $$ _#=_4
6. Parser_3 && Bigram_2 && Synset_2 && BagOfWord_2 $$ _3=_2
7. Parser_3 && Bigram $$ _3=_1
8. Parser_0 $$ _#=_0
……

Rule 1 shows that: in cases where Dependency Structure is adopted as the feature, when the classification result is 1 and the question is not classified, the question belongs to the first class. Rule 2, 3, 4, and 5 is similar to 1.

Rules 6, 7 involve classification results from multiple classifiers. Rule 6 indicates that, if sentence is placed in 3 when Dependency Structure is adopted as feature, in class 2, when Bi-gram or Synset or Bag-of-Word is adopted, and questions have already been tagged as 3, it will be put in class 2.

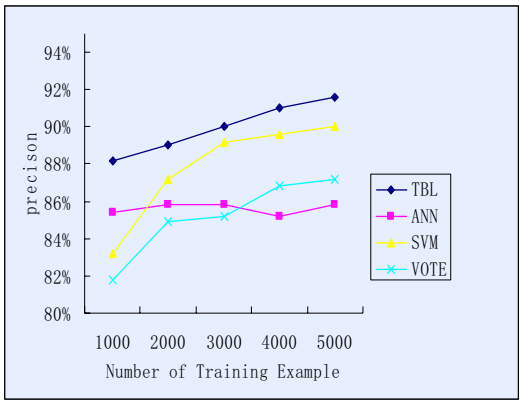

Figure 6. Different combine method

Figure 6 gives us the classification results of 500 questions of Trec10 in different method of combination. It can be seen that, TBL combination of classifiers is better than voting and ANN; TBL and SVM working together is better than SVM classification using multi-type-features to represent questions directly.

Figure 7. Using dependency structure or not

Figure7 provides a comparison of classification accuracy between TBL combining multi-classifier and SVM directly using several type features, in conditions of adopting or not adopting Dependency Structure as feature. TBL- and SVM- both mean classifier not adopting. The results show: Using such method of QC as blending "statistics" and "rules", that is, the accuracy of classification is 1.6% greater than that of not using TBL; adopting Dependency Structure as feature can also promote precision, with a percentage of 1.8.

4.7 Result Analysis

Compared to (Zhang, et al. 2003) using "tree kernel" as the classification feature, this thesis adopts the "statistics and rules blended" method in QC ("statistics first and rules next"), lifting the precision of classification to 1.4% higher than it used to be. Moreover, it also avoids the problem of artificial selection in different feature weighting that appearing in Zhang's paper.

Tests using the "statistics and rules blended" pattern of question classification unfold that, 34.1% of faulty classification of sentences arouses from the using of improper statistical methods. The manifest of this is that all the SVM classifiers with 4 features place questions into class *"i"*, while they actually belong to class *"j"*. Classification features that have relatively big differences are needed to

work as basic classifier to improve the final result. And also, 31.8% of the faulty classification stems from the fact that, there are no corresponding rules in the rule sets derived from TBL training, so that the rule sets cannot correct the errors caused by wrong statistical methods. This may because our question corpus is Limited, and therefore, some of the classification combinations never even appear.

5 Conclusions

QC, an important module in the QA system, can conduct answer choosing and selection. This thesis experiment several different methods in QC, and studies features like the Dependency Structure, Wordnet Synsets, Bag-of-Word, and Bi-gram. It also analyzes a number of kernel functions and the influence of different ways of classifier combination, such as Voting, Adaboost、ANN and TBL, on the precision of QC. Adopting the “statistics and rules blended” method of question classification (“statistics first and rules next”) and using language information such as the Synset from Wordnet and the dependency structure of Minipar as classification features promote the accuracy of question classification. TBL combination multi-classifier method can be extended, easily. As long as new classifying algorithm or new feature set is found, the classifying result from them can be transformed to rule set, which can lead to further classifying function. Wordnet has provided us with semantic relation, examples, explanation, etc. The present study only investigates the semantic relation of hyponymy. There are still much to be done in the future to further the research on QC using Wordnet.

Reference

Brill, E. 1995. "*Transformation-based error-driven learning and natural language processing: a case study part-of-speech tagging*." *Computational Linguistics* 2:543-565.

Chang, C.-C. and Lin, C.-J. 2001. " *LIBSVM: a library for support vector machines*." available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm

Hovy, E., Hermjakob, U., Lin, C.-Y. and Ravichandran, D. 2002. "*Using Knowledge to Facilitate Factoid Answer Pinpointing*." In, *Proceedings of the COLING-2002 Conference.* Taipei, Taiwan.

Ittycheriah, A., Franz, M., Zhu, W.-J. and Ratnaparkhi, A. 2000. "*IBM's Statistical Question Answering System*." In, *Proceedings of the TREC-9 Conference.* Gaithersburg, MD: NIST, p. 229.

Li, X. and Roth, D. 2002. "*Learning Question Classifiers*." In, *Proceedings of the 19th International Conference on Computational Linguistics, (COLING'02).* Taipei.

Lin, D. 1998. "*Dependency-based Evaluation of MINIPAR*." *In Workshop on the Evaluation of Parsing Systems.* Granada, Spain.

Miller, G. 1995. "*WordNet: A Lexical Database for English*." *Communications of the ACM* 38(11):39-41.

Miller, G. and McDonnell, J. S. 2003. "*Wordnet 2.0*." Princeton University's Cognitive Science Laboratory.

Moldovan, D., PASCA, M. and HARABAGIU, S. 2003. "*Performance Issues and Error Analysis in an Open-Domain Question Answering System*." *ACM Transactions on Information Systems* Vol.21:133-154.

Schapire, R. E. 1997. "*Using output codes to boost multiclass learning problems*." *In Machine Learning: Proceedings of the Fourteenth International Conference.* pp. 313-321.

Schapire, R. E. 1999. "*Theoretical views of boosting and applications*." *In 10th International Conference on Algorithmic Learning Theory.*

Zhang, D. and Lee, W. S. 2003. "*Question Classification using Support Vector Machines*." In, *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03).* Toronto, Canada: ACM.