Investigation of Question Classifier in Question Answering

Zhiheng Huang
EECS Department
University of California
at Berkeley
CA 94720-1776, USA
zhiheng@cs.berkeley.edu

Marcus Thint
Intelligent Systems Research Center
British Telecom Group
Chief Technology Office
marcus.2.thint@bt.com

Asli Celikyilmaz
EECS Department
University of California
at Berkeley
CA 94720-1776, USA
asli@cs.berkeley.edu

Abstract

In this paper, we investigate how an accurate question classifier contributes to a question answering system. We first present a Maximum Entropy (ME) based question classifier which makes use of head word features and their WordNet hypernyms. We show that our question classifier can achieve the state of the art performance in the standard UIUC question dataset. We then investigate quantitatively the contribution of this question classifier to a feature driven question answering system. With our accurate question classifier and some standard question answer features, our question answering system performs close to the state of the art using TREC corpus.

1 Introduction

Question answering has drawn significant attention from the last decade (Prager, 2006). It attempts to answer the question posed in natural language by providing the answer phrase rather than the whole documents. An important step in question answering (QA) is to classify the question to the anticipated type of the answer. For example, the question of *Who discovered x-rays* should be classified into the type of human (individual). This information would narrow down the search space to identify the correct answer string. In addition, this information can suggest different strategies to search and verify a candidate answer. In fact, the combination of question classification and the named entity recognition is a key approach in modern question answering systems (Voorhees and Dang, 2005).

The question classification is by no means trivial: Simply using question wh-words can not achieve satisfactory results. The difficulty lies in classifying the *what* and *which* type questions. Considering the example *What is the capital of Yugoslavia*, it is of location (city) type, while *What is the pH scale* is of definition type. As with the previous work of (Li and Roth, 2002; Li and Roth, 2006; Krishnan et al., 2005; Moschitti et al., 2007), we propose a feature driven statistical question classifier (Huang et al., 2008). In particular, we propose head word feature and augment semantic features of such head words using WordNet. In addition, Lesk's word sense disambiguation (WSD) algorithm is adapted and the depth of hypernym feature is optimized. With further augment of other standard features such as unigrams, we can obtain accuracy of 89.0% using ME model for 50 fine classes over UIUC dataset.

In addition to building an accurate question classifier, we investigate the contribution of this question classifier to a feature driven question answering rank model. It is worth noting that, most of the features we used in question answering rank model, depend on the question type information. For instance, if a question is classified as a type of *sport*, we then only care about whether there are sport entities existing in the candidate sentences. It is expected that a fine grained named entity recognizer (NER) should make good use of the accurate question type information. However, due to the lack of a fine grained NER tool at hand, we employ the Stanford NER package (Finkel et al., 2005) which identifies only four types of named entities. Even with such a coarse named entity recognizer, the experiments show that the question classifier plays an important role in determining the performance of a question answering system.

The rest of the paper is organized as following. Section 2 reviews the maximum entropy model which are used in both question classification and question answering ranking. Section 3 presents the features used in question classification. Section 4 presents the question classification

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 543–550,
Singapore, 6-7 August 2009.

accuracy over UIUC question dataset. Section 5 presents the question answer features. Section 6 illustrates the results based on TREC question answer dataset. And Section 7 draws the conclusion.

2 Maximum Entropy Models

Maximum entropy (ME) models (Berger et al., 1996; Manning and Klein, 2003), also known as log-linear and exponential learning models, provide a general purpose machine learning technique for classification and prediction which has been successfully applied to natural language processing including part of speech tagging, named entity recognition etc. Maximum entropy models can integrate features from many heterogeneous information sources for classification. Each feature corresponds to a constraint on the model. Given a training set of (C, D), where C is a set of class labels and D is a set of feature represented data points, the maximal entropy model attempts to maximize the log likelihood

$$\log P(C|D,\lambda) = \sum_{(c,d)\in(C,D)} \log \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_j \lambda_j f_i(c,d)}, \quad (1)$$

where $f_i(c,d)$ are feature indicator functions. We use ME models for both question classification and question answer ranking. In question answer context, such function, for instance, could be the presence or absence of dictionary entities (as presented in Section 5.2) associated with a particular class type (either *true* or *false*, indicating a sentence can or cannot answer the question). λ_i are the parameters need to be estimated which reflects the importance of $f_i(c,d)$ in prediction.

3 Question Classification Features

Li and Roth (2002) have developed a machine learning approach which uses the SNoW learning architecture. They have compiled the UIUC question classification dataset[1] which consists of 5500 training and 500 test questions.[2] All questions in the dataset have been manually labeled according to the coarse and fine grained categories as shown in Table 1, with coarse classes (in bold) followed by their fine classes.

The UIUC dataset has laid a platform for the follow-up research including (Hacioglu and Ward, 2003; Zhang and Lee, 2003; Li and Roth, 2006; Krishnan et al., 2005; Moschitti et al., 2007). In contrast to Li and Roth (2006)'s approach which makes use of a very rich feature set, we propose to use a compact yet effective feature set. The features are briefly described as following. More detailed information can be found at (Huang et al., 2008).

Table 1: 6 coarse and 50 fine Question types defined in UIUC question dataset.

ABBR	letter	desc	**NUM**
abb	other	manner	code
exp	plant	reason	count
ENTITY	product	**HUMAN**	date
animal	religion	group	distance
body	sport	individual	money
color	substance	title	order
creative	symbol	desc	other
currency	technique	**LOC**	period
dis.med.	term	city	percent
event	vehicle	country	speed
food	word	mountain	temp
instrument	**DESC**	other	size
lang	definition	state	weight

Question wh-word The wh-word feature is the question wh-word in given questions. For example, the wh-word of question *What is the population of China* is *what*.

Head Word *head word* is defined as one *single* word specifying the object that the question seeks. For example the head word of *What is a group of turkeys called*, is *turkeys*. This is different to previous work including (Li and Roth, 2002; Krishnan et al., 2005) which has suggested a contiguous span of words (*a group of turkeys* in this example). The single word definition effectively avoids the noisy information brought by non-head word of the span (*group* in this case). A syntactic parser (Petrov and Klein, 2007) and the Collins rules (Collins, 1999) are modified to extract such head words.

WordNet Hypernym WordNet hypernyms are extracted for the head word of a given question. The classic Lesk algorithm (Lesk, 1986) is used to compute the most probable sense for a head word in the question context, and then the hypernyms are extracted based on that sense. The depth of hypernyms is set to

[1] Available at http://12r.cs.uiuc.edu/∼cogcomp/Data/QA/QC.
[2] Test questions are from TREC 10.

six with trial and error.[3] Hypernyms features capture the general terms of extracted head word. For instance, the head word of question *What is the proper name for a female walrus* is extracted as *walrus* and its direct hypernyms such as *mammal* and *animal* are extracted as informative features to predict the correct question type of ENTY:animal.

Unigram words Bag of words features. Such features provide useful question context information.

Word shape Five word shape features, namely all upper case, all lower case, mixed case, all digits, and other are used to serve as a coarse named entity recognizer.

4 Question Classification Experiments

We train a Maximum Entropy model using the UIUC 5500 training questions and test over the 500 test questions. Tables 2 shows the accuracy of 6 coarse class and 50 fine grained class, with features being fed incrementally. The question classification performance is measured by accuracy, i.e., the proportion of the correctly classified questions among all test questions. The baseline using the wh-head word results in 46.0% and 46.8% respectively for 6 coarse and 50 fine class classification. The incremental use of head word boosts the accuracy significantly to 92.2% and 82.0% for 6 and 50 classes. This reflects the informativeness of such feature. The inclusion of hypernym feature within 6 depths boosts 3.6% for 50 classes, while resulting in slight loss for 6 coarse classes. The further use of unigram feature leads to 2.8% gain in 50 classes. Finally, the use of word shape leads to 0.6% accuracy increase for 50 classes. The best accuracies achieved are 93.6% and 89.0% for 6 and 50 classes respectively.

Table 2: Question classification accuracy using incremental feature sets for 6 and 50 classes over UIUC split.

	6 class	50 class
wh-word	46.0	46.8
+ head word	92.2	82.0
+ hypernym	91.8	85.6
+ unigram	93.0	88.4
+ word shape	93.6	89.0

The individual feature contributions were discussed in greater detail in (Huang et al., 2008). Also, The SVM (rathern than ME model) was employed using the same feature set and the results were very close (93.4% for 6 class and 89.2% for 50 class). Table 3 shows the feature ablation experiment[4] which is missing in that paper. The experiment shows that the proposed head word and its hypernym features play an essential role in building an accurate question classifier.

Table 3: Question classification accuracy by removing one feature at a time for 6 and 50 classes over UIUC split.

	6 class	50 class
overall	93.6	89.0
- wh-word	93.6	89.0
- head word	92.8	88.2
- hypernym	90.8	84.2
- unigram	93.6	86.8
- word shape	93.0	88.4

Our best result feature space only consists of 13'697 binary features and each question has 10 to 30 active features. Compared to the over feature size of 200'000 in Li and Roth (2002), our feature space is much more compact, yet turned out to be more informative as suggested by the experiments. Table 4 shows the summary of the classification accuracy of all question classifiers which were applied to UIUC dataset.[5] Our results are summarized in the last row.

In addition, we have performed the 10 cross validation experiment over the 5500 UIUC training corpus using our best model. The result is 89.05 ± 1.25 and 83.73 ± 1.61 for 6 and 50 classes,[6] which outperforms the best result of 86.1 ± 1.1 for 6 classes as reported in (Moschitti et al., 2007).

5 Question Answer Features

For a pair of a question and a candidate sentence, we extract binary features which include CoNLL named entities presence feature (NE), dictionary

[3] We performed 10 cross validation experiment over training data and tried various depths of 1, 3, 6, 9 and ∞, with ∞ signifies that no depth constraint is imposed.

[4] Remove one feature at a time from the entire feature set.

[5] Note (1) that SNoW accuracy without the related word dictionary was not reported. With the semantically related word dictionary, it achieved 91%. Note (2) that SNoW with a semantically related word dictionary achieved 84.2% but the other algorithms did not use it.

[6] These results are worse than the result over UIUC split; as the UIUC test data includes a larger percentage of easily classified question types.

Table 4: Accuracy of all question classifiers which were applied to UIUC dataset.

Algorithm	6 class	50 class
Li and Roth, SNoW	$-^{(1)}$	$78.8^{(2)}$
Hacioglu et al., SVM+ECOC	–	80.2-82
Zhang & Lee, Linear SVM	87.4	79.2
Zhang & Lee, Tree SVM	90.0	–
Krishnan et al., SVM+CRF	93.4	86.2
Moschitti et al., Kernel	91.8	–
Maximum Entropy Model	93.6	89.0

entities presence feature (DIC), numerical entities presence feature (NUM), question specific feature (SPE), and dependency validity feature (DEP).

5.1 CoNLL named entities presence feature

We use Stanford named entity recognizer (NER) (Finkel et al., 2005) to identify CoNLL style NEs[7] as possible answer strings in a candidate sentence for a given type of question. In particular, if the question is ABBR type, we tag CoNLL LOC, ORG and MISC entities as candidate answers; If the question is HUMAN type, we tag CoNLL PER and ORG entities; And if the question is LOC type, we tag CoNLL LOC and MISC entities. For other types of questions, we assume there is no candidate CoNLL NEs to tag. We create a binary feature **NE** to indicate the presence or absence of tagged CoNLL entities. Further more, we create four binary features **NE-PER**, **NE-LOC**, **NE-ORG**, and **NE-MISC** to indicate the presence of tagged CoNLL PER, LOC, ORG and MISC entities.

5.2 Dictionary entities presence feature

As four types of CoNLL named entities are not enough to cover 50 question types, we include the 101 dictionary files compiled in the Ephyra project (Schlaefer et al., 2007). These dictionary files contain names for specific semantic types. For example, the *actor* dictionary comprises a list of actor names such as *Tom Hanks* and *Kevin Spacey*. For each question, if the head word of such question (see Section 3) matches the name of a dictionary file, then each noun phrase in a candidate sentence is looked up to check its presence in the dictionary. If so, a binary **DIC** feature is created. For example, for the question *What rank did Chester Nimitz reach*, as there is a *military rank* dictionary matches the head word *rank*, then all the noun phrases in a candidate sentence are looked up in the *military rank* dictionary. As a result, a sentence contains word *Admiral* will result in the DIC feature being activated, as such word is present in the *military rank* dictionary.

Note that an implementation tip is to allow the proximity match in the dictionary look up. Consider the question *What film introduced Jar Jar Binks*. As there is a match between the question head word *film* and the dictionary named *film*, each noun phrase in the candidate sentence is checked. However, no dictionary entities have been found from the candidate sentence *Best plays Jar Jar Binks, a floppy-eared, two-legged creature in "Star Wars: Episode I – The Phantom Menace"*, although there is movie entitled *Star Wars Episode I: The Phantom Menace* in the dictionary. Notice that *Star Wars: Episode I – The Phantom Menace* in the sentence and the dictionary entity *Star Wars Episode I: The Phantom Menace* do not have exactly identical spelling. The use of proximity look up which allows edit distance being less than 10% error can resolve this.

5.3 Numerical entities presence feature

There are so far no match for question types of NUM (as shown in Table 1) including NUM:count and NUM:date etc. These types of questions seek the numerical answers such as the amount of money and the duration of period. It is natural to compile regular expression patterns to match such entities. For example, for a NUM:money typed question *What is Rohm and Haas's annual revenue*, we compile NUM:money regular expression pattern which matches the strings of number followed by a currency sign (*$* and *dollars* etc). Such pattern is able to identify *4 billion $* as a candidate answer in the candidate sentence *Rohm and Haas, with 4 billion $ in annual sales...* There are 13 patterns compiled to cover all numerical types. We create a binary feature **NUM** to indicate the presence of possible numerical answers in a sentence.

5.4 Specific features

Specific features are question dependent. For example, for question *When was James Dean born*, any candidate sentence matches the pattern *James Dean (number - number)* is likely to answer such question. We create a binary feature **SPE** to indicate the presence of such match between a ques-

[7] Person (PER), location (LOC), organization (ORG), and miscellaneous (MISC).

tion and a candidate sentence. We list all question and sentence match patterns which are used in our experiments as following:

when born feature 1 The question begins with *when is/was* and follows by a person name and then follows by key word *born*; The candidate sentence contains such person name which follows by the pattern of *(number - number)*.

when born feature 2 The question begins with *when is/was* and follows by a person name and then follows by key word *born*; The candidate sentence contains such person name, a NUM:date entity, and a key word *born*.

where born feature 1 The question begins with *where is/was* and follows by a person name and then follows by key word *born*; The candidate sentence contains such person name, a NER LOC entity, and a key word *born*.

when die feature 1 The question begins with *when did* and follows by a person name and then follows by key word *die*; The candidate sentence contains such person name which follows by the pattern of *(number - number)*.

when die feature 2 The question begins with *when did* and follows by a person name and then follows by key word *die*; The candidate sentence contains such person name, a NUM:date entity, and a key word *died*.

how many feature The question begins with *how many* and follows by a noun; The candidate sentence contains a number and then follows by such noun.

cooccurrent Feature This feature takes two phrase arguments, if the question contains the first phrase and the candidate sentence contains the second, such feature would be activated.

Note that the construction of specific features require the access to aforementioned extracted named entities. For example, the **when born feature 2** pattern needs the information whether a candidate sentence contains a NUM:date entity and **where born feature 1** pattern needs the information whether a candidate sentence contains a NER LOC entity. Note also that the patterns of **when born feature** and **when die feature** have similar structure and thus can be simplified in implementation. **How many feature** can be used to identify the sentence *Amtrak annually serves about 21 million passengers* for question *How many passengers does Amtrak serve annually*. The **cooccurrent feature** is the most general one. An example of cooccurrent feature would take the arguments of *marry* and *husband*, or *marry* and *wife*. Such feature would be activated for question *Whom did Eileen Marie Collins marry* and candidate sentence *... were Collins' husband, Pat Youngs, an airline pilot...* It is worth noting that the two arguments are not necessarily different. For example, they could be both *established*, which makes such feature activated for question *When was the IFC established* and candidate sentence *IFC was established in 1956 as a member of the World Bank Group*. The reason why we use the cooccurrence of the word *established* is due to its main verb role, which may carry more information than other words.

5.5 Dependency validity features

Like (Cui et al., 2004), we extract the dependency path from the question word to the common word (existing in both question and sentence), and the path from candidate answer (such as CoNLL NE and numerical entity) to the common word for each pair of question and candidate sentence using Stanford dependency parser (Klein and Manning, 2003; Marneffe et al., 2006). For example, for question *When did James Dean die* and candidate sentence *In 1955, actor James Dean was killed in a two-car collision near Cholame, Calif.*, we extract the pathes of *When:advmod:nsubj:Dean* and *1955:prep-in:nsubjpass:Dean* for question and sentence respectively, where *advmod* and *nsubj* etc. are grammatical relations. We propose the dependency validity feature (**DEP**) as following. For all paired paths between a question and a candidate sentence, if at least one pair of path in which all pairs of grammatical relations have been seen in the training, then the DEP feature is set to be true, false otherwise. That is, the true validity feature indicates that at least one pair of path between the question and candidate sentence is possible to be a true pair (ie, the candidate noun phrase in the sentence path is the true answer).

6 Question Answer Experiments

Recall that most of the question answer features depend on the question classifier. For instance, the NE feature checks the presence or absence of CoNLL style named entities subject to the classified question type. In this section, we evaluate how the quality of question classifiers affects the question answering performance.

6.1 Experiment setup

We use TREC99-03 factoid questions for training and TREC04 factoid questions for testing. To facilitate the comparison to others work (Cui et al., 2004; Shen and Klakow, 2006), we first retrieve all relevant documents which are compiled by Ken Litkowski[8] to create training and test datasets. We

[8] Available at http://trec.nist.gov/data/qa.html.

then apply key word search for each question and retrieve the top 20 relevant sentences. We create a feature represented data point using each pair of question and candidate sentence and label it either *true* or *false* depending on whether the sentence can answer the given question or not. The labeling is conducted by matching the gold factoid answer pattern against the candidate sentence.

There are two extra steps performed for training set but not for test data. In order to construct a high quality training set, we manually check the correctness of the training data points and remove the false positive ones which cannot support the question although there is a match to gold answer. In addition, in order to keep the training data well balanced, we keep maximum four false data points (question answer pair) for each question but no limit over the true label data points. In doing so, we use 1458 questions to compile 8712 training data points and among them 1752 have true labels. Similarly, we use 202 questions to compile 4008 test data points and among them 617 have true labels.

We use the training data to train a maximum entropy model and use such model to rank test data set. Compared with a classification task (such as the question classifier), the ranking process requires one extra step: For data points which share the same question, the probabilities of being predicted as true label are used to rank the data points. In align with the previous work, performance is evaluated using mean reciprocal rank (MRR), top 1 prediction accuracy (top1) and top 5 prediction accuracy (top5). For the test data set, 157 among the 202 questions have correct answers found in retrieved sentences. This leads to the upper bound of MRR score being 77.8%.

To evaluate how the quality of question classifiers affects the question answering, we have created three question classifiers: QC1, QC2 and QC3. The features which are used to train these question classifiers and their performance are shown in Table 5. Note that QC3 is the best question classifier we obtained in Section 4.

Table 5: Features used to train and the performance of three question classifiers.

Name	features	6 class	50 class
QC1	wh-word	46.0	46.8
QC2	wh-word+ head	92.2	82.0
QC3	All	93.6	89.0

6.2 Experiment results

The first experiment is to evaluate the individual contribution of various features derived using three question classifiers. Table 6 shows the baseline result and results using DIC, NE, NE-4, REG, SPE, and DEP features. The baseline is the key word search without the use of maximum entropy model. As can be seen, the question classifiers do not affect the DIC feature at all, as DIC feature does not depend on question classifiers. Better question classifier boosts considerable gain for NE, NE-4 and REG in their contribution to question answering. For example, the best question classifier QC3 outperforms the worst one (QC1) by 1.5%, 2.0%, and 2.0% MRR scores for NE, NE-4 and REG respectively. However, it is surprising that the MRR and top5 contribution of NE and NE-4 decreases if QC1 is replaced by QC2, although the top1 score results in performance gain slightly. This unexpected results can be partially explained as follows. For some questions, even QC2 produces correct predictions, the errors of NE and NE-4 features may cause over-confident scores for certain candidate sentences. As SPE and DEP are not directly dependent on question classifier, their individual contribution only changes slightly or remains the same for different question classifiers. If the best question classifier is used, the most important features are SPE and REG, which can individually boost the MRR score over 54%, while the others result in less significant gains.

We now incrementally use various features and the results are show in Table 6 as well. As can be seen, the more features and the better question classifier are used, the higher performance the ME model has. The inclusion of REG and SPE results in significant boost for the performance. For example, if the best question classifier QC3 is used, the REG results in 6.9% and 8% gain for MRR and top1 scores respectively. This is due to a large portion of NUM type questions in test dataset. The SPE feature contributes significantly to the performance due to its high precision in answering birth/death time/location questions. NE and NE-4 result in reasonable gains while DEP feature contributes little. However, this does not mean that DEP is not important, as once the model reaches a high MRR score, it becomes hard to improve.

Table 6 clearly shows that the question type classifier plays an essential role in a high perfor-

Table 6: Performance of individual and incremental feature sets for three question classifiers.

Feature	MRR			Top1			Top5		
	QC1	QC2	QC3	QC1	QC2	QC3	QC1	QC2	QC3
Individual									
Baseline	49.9	49.9	49.9	40.1	40.1	40.1	59.4	59.4	59.4
DIC	49.5	49.5	49.5	42.6	42.6	42.6	60.4	60.4	60.4
NE	48.5	47.5	50.0	40.6	40.6	42.6	61.9	60.9	63.4
NE-4	49.5	48.5	51.5	41.6	42.1	44.6	62.4	61.9	64.4
REG	52.0	54.0	54.0	44.1	47.0	47.5	64.4	65.3	65.3
SPE	55.0	55.0	55.0	48.5	48.5	48.5	64.4	64.4	64.4
DEP	51.0	51.5	52.0	43.6	44.1	44.6	65.3	65.8	65.8
Incremental									
Baseline	49.9	49.9	49.9	40.1	40.1	40.1	59.4	59.4	59.4
+DIC	49.5	49.5	49.5	42.6	42.6	42.6	60.4	60.4	60.4
+NE	50.0	48.5	51.0	43.1	42.1	44.6	62.9	61.4	64.4
+NE-4	51.5	50.0	53.0	44.1	43.6	46.0	63.4	62.9	65.8
+REG	55.0	56.9	59.9	48.0	51.0	54.0	68.3	68.8	71.8
+SPE	60.4	62.4	65.3	55.4	58.4	61.4	70.8	70.8	73.8
+DEP	61.4	62.9	**66.3**	55.9	58.4	**62.4**	71.8	71.8	**73.8**

mance question answer system. Assume all the features are used, the better question classifier significantly boosts the overall performance. For example, the best question classifier QC3 outperforms the worst QC1 by 4.9%, 6.5%, and 2.0% for MRR, top1 and top5 scores respectively. Even compared to a good question classifier QC2, the gain of using QC3 is still 3.4%, 4.0% and 2.0% for MRR, top1 and top5 scores respectively. One can imagine that if a fine grained NER is available (rather than the current four type coarse NER), the potential gain is much significant.

The reason that the question classifier affects the question answering performance is straightforward. As a upstream source, the incorrect classification of question type would confuse the downstream answer search process. For example, for question *What is Rohm and Haas's annual revenue*, our best question classifier is able to classify it into the correct type of NUM:money and thus would put *$ 4 billion* as a candidate answer. However, the inferior question classifiers misclassify it into HUM:ind type and thereby could not return a correct answer. Figure 1 shows the individual MRR scores for the 42 questions (among the 202 test questions) which have different predicted question types using QC3 and QC2. For almost all test questions, the accurate question classifier QC3 achieves higher MRR scores compared to QC2.

Table 7 shows performance of various question answer systems including (Tanev et al., 2004; Wu et al., 2005; Cui et al., 2004; Shen and Klakow, 2006) and this work which were applied to the same training and test datasets. Among all the systems, our model can achieve the best MRR score of 66.3%, which is close to the state of the art of 67.0%. Considering the question answer features used in this paper are quite standard, the boost is mainly due to our accurate question classifier.

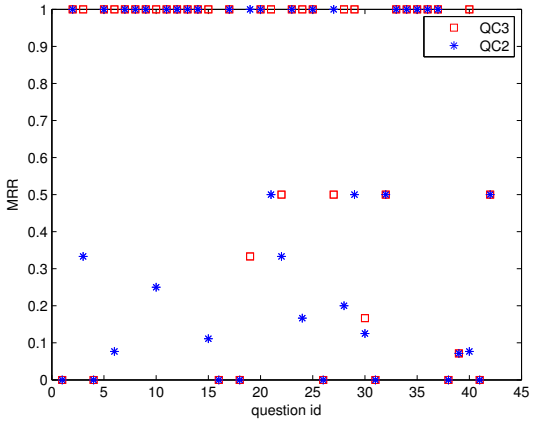

Figure 1: Individual MRR scores for questions which have different predicted question types using QC3 and QC2.

Table 7: Various system performance comparison.

System	MRR	Top1	Top5
Tanev et al. 2004	57.0	49.0	67.0
Cui et al. 2004	60.0	53.0	70.0
Shen and Klakow, 2006	67.0	62.0	74.0
This work	66.3	62.4	73.8

7 Conclusion

In this paper, we have presented a question classifier which makes use of a compact yet efficient feature set. The question classifier outperforms previous question classifiers over the standard UIUC question dataset. We further investigated quantitatively how the quality of question classifier impacts the performance of question answer system. The experiments showed that an accurate question classifier plays an essential role in question answering system. With our accurate question classifier and some standard question answer features, our question answering system performs close to the state of the art.

Acknowledgments

We wish to thank the three anonymous reviewers for their invaluable comments. This research was supported by British Telecom grant CT1080028046 and BISC Program of UC Berkeley.

References

A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

M. Collins. 1999. Head-driven statistical models for natural language parsing. *PhD thesis*, University of Pennsylvania.

H. Cui, K Li, R. Sun, T. Chua, and M. Kan. 2004. National university of singapore at the trec-13 question answering. In *Proc. of TREC 2004, NIST*.

J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of ACL*, pages 363-370.

D. Hacioglu and W. Ward. 2003. Question classification with support vector machines and error correcting codes. In *Proc. of the ACL/HLT*, vol. 2, pages 28–30.

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

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL 2003*, vol. 1, pages 423–430.

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

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *ACM Special Interest Group for Design of Communication Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26.

X. Li and D. Roth. 2002. Learning question classifiers. In *the 19th international conference on Computational linguistics*, vol. 1, pages 1-7.

X. Li and D. Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249.

C. D. Manning and D. Klein. 2003. Optimization, maxent models, and conditional estimation without magic. *Tutorial at HLT-NAACL 2003 and ACL 2003*.

M. D. Marneffe, B. MacCartney and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC 2006*.

A. Moschitti, S. Quarteroni, R. Basili and S. Manandhar 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proc. of ACL 2007*, pages 776-783.

S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In *Proc. of the HLT-NAACL.*

J. Prager. 2006. Open-domain question-answering. In *Foundations and Trends in Information Retrieval*, vol. 1, pages 91-231, 2006.

N. Schlaefer, J. Ko, J. Betteridge, G. Sautter, M. Pathak and E. Nyberg. 2007. Semantic extensions of the Ephyra QA system for TREC 2007. In *Proc. of the TREC 2007*.

D. Shen and D. Klakow. 2006. Exploring correlation of dependency relation paths for answer extraction. In *Proc. of the ACL 2006*.

H. Tanev, M. Kouylekov, and B. Magnini. 2004. Combining linguistic processing and web mining for question answering: Itc-irst at TREC-2004. In *Proc. of the TREC 2004, NIST*.

E. M. Voorhees and H. T. Dang. 2005. Overview of the TREC 2005 question answering track. In *Proc. of the TREC 2005, NIST*.

M. Wu, M. Duan, S. Shaikh, S. Small, and T. Strzalkowski. 2005. University at Albanys ILQUA in TREC 2005. In *Proc. of the TREC 2005, NIST*.

D. Zhang and W. S. Lee. 2003. Question classification using support vector machines. In *The ACM SIGIR conference in information retrieval*, pages 26–32.