Classifying *What-type* Questions by *Head Noun* Tagging

Fangtao Li, Xian Zhang, Jinhui Yuan, Xiaoyan Zhu
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Sci. and Tech., Tsinghua University, Beijing 100084, China
zxy-dcs@tsinghua.edu.cn

Abstract

Classifying *what-type* questions into proper semantic categories is found more challenging than classifying other types in question answering systems. In this paper, we propose to classify *what-type* questions by *head noun* tagging. The approach highlights the role of *head nouns* as the category discriminator of *what-type* questions. To reduce the semantic ambiguities of *head noun*, we integrate local syntactic feature, semantic feature and category dependency among adjacent nouns with Conditional Random Fields (CRFs). Experiments on standard question classification data set show that the approach achieves state-of-the-art performances.

1 Introduction

Question classification is a crucial component of modern question answering system. It classifies questions into several semantic categories which indicate the expected semantic type of answers to the questions. The semantic category helps to filter out irrelevant answer candidates, and determine the answer selection strategies.

The widely used question category criteria is a two-layered taxonomy developed by Li and Roth (2002) from UIUC. The hierarchy contains 6 coarse classes and 50 fine classes as shown in Table 1. In this paper, we focus on fine-category classification. Each fine category will be denoted as "Coarse:fine", such as "HUM:individual".

A *what-type* question is defined as the one whose question word is "what", "which", "name" or "list". It is a dominant type in question answering system. Li and Roth (2006) find that the distribution of *what-type* questions over the semantic classes is quite diverse, and they are more difficult to be classified than other types. Table 2 shows the classification accuracies of each question word in UIUC data set using Support Vector Machine (SVM) with unigram features. *What-type* questions account for more than 70 percent in the data set, but the classification accuracy of this type only achieves 75.50%. In this experiment, 90.53% (86 over 95) of the errors are generated by *what-type* questions. Due to its challenge, this paper focuses on *what-type* question classification.

Coarse	**Fine**
ABBR	abbreviation, expression
DESC	definition, description, manner, reason
ENTY	animal, body, color, creation, currency, disease/medicine, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
HUM	description, group, individual, title
LOC	city, country, mountain, other, state
NUM	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight

Table 1. Question Ontology

	Total	Wrong	Accuracy
What-type	351	86	75.50%
Where	26	2	92.31%
When	26	0	100.0%
Who	47	3	93.62%
How	46	4	91.30%
Why	4	0	100.0%
Total	500	95	81.00%

Table 2. Classification performance for each question words with unigram

Head noun has been presented to play an important role in classifying *what-type* questions (Metzler and Croft, 2005). It refers to the noun reflecting the focus of a question, such as "flower" in the question "What is Hawaii's state

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 481–488
Manchester, August 2008

flower?". These nouns can effectively reduce the noise generated by other words. If the *head noun* "length" is identified from the question "What is the length of the coastline of the state of Alaska?", this question can be easily classified into "NUM:distance". However, the above SVM misclassified this question into "LOC:-state", as the words "state" and "Alaska" confused the classifier. Considering another two questions expressed in (Zhang and Lee, 2002), "Which university did the president graduate from?" and "Which president is a graduate of the Harvard University", although they contain similar words, it is not difficult to distinguish them with the *head nouns* "university" and "president" respectively.

Nevertheless, a *head noun* may correspond to several semantic categories. In this situation, we need to incorporate the *head noun* context for disambiguation. The potentially useful context features include local syntactic features, semantic features and neighbor's semantic category. Take the noun "money" as an example, it possibly corresponds to two categories: "NUM:money" and "ENTY:currency". If there is an adjacent word falling into "Loc:country" category, the "money" tends to belong to "ENTY:currency". Otherwise, if the "ENTY:product" or "HUM:individual" surrounds it, the word "money" may refer to "NUM:money".

Based on the above notions, we propose a new strategy to classify *what-type* questions by word tagging, and the selected *head noun* determines question category. The question classification task is formulated into word sequence tagging problem. All the question words are divided into semantic words and non-semantic words. The semantic word expresses certain semantic category, such as "dog" corresponding to category "ENTITY:animal", while "have" corresponding to no category. The label for semantic words is one of the question categories, and "O" is for non-semantic word. Here, we just consider the nouns as semantic words, others as non-semantic words. Each word in a question will be tagged as a label using Conditional Random Fields model, and the *head noun*'s label is chosen as the question category.

In conclusion, the CRFs based approach has two main steps: the first step is to tag all the words in questions using CRFs, and the second step is choosing the *head noun*'s label as the question category. It can use the *head noun* to eliminate the noisy words, and take advantages of CRFs model to integrate not only the syntactic and semantic features, but also the adjacent categories to tag *head noun*.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 introduces the Condition Random Fields(CRFs) and the defined Long-Dependency CRFs (LDCRFs). Section 4 describes the features used in the LDCRFs. The *head noun* extraction method is presented in Section 5. We evaluate the proposed approach in Section 6. Section 7 concludes this paper and discusses future work.

2 Related works

Question Answering Track was first introduced in the Text REtrieval Conference (TREC) in 1999. Since then, question classification has been a popular topic in the research community of text mining. Simple question classification approaches usually employ hand-crafted rules (such as Prager et. al, 1999), which are effective for specific question taxonomy. However, laborious human effort is required to create these rules.

Some other systems employed machine learning approaches to classify questions. Li and Roth (2002) presented a hierarchical classifier based on the Sparse Network of Winnows (Snow) architecture. Tow classifiers were involved in this work: the first one classified questions into the coarse categories; and the other classified questions into fine categories. Several syntactic and semantic features, including semi-automatically constructed class-specific relational features, were extracted and compared in their experiments. The results showed that the hierarchical classifier was effective for question classification task.

Metzler and Croft (2005) used prior knowledge about correlations between question words and types to train word-specific question classifiers. They identified the question words firstly, and trained separate classifier for each question word. WordNet was used as semantic features to boost the classification performance. In this paper, according to question word, all the questions are classifie into two categories: *what-type* ones and non-*what-type* one.

Recent question classification methods have paid more attention on the syntactic structure of sentence. They used a parser to get the syntactic tree, and then took advantage of the structure information. Zhang and Lee (2002) proposed a tree kernel Support Vector Machine classifier and experiment results showed that syntactic information and tree kernel could solve this prob-

lem. Nguyen et al. (2007) proposed a subtree mining method for question classification. They formulated question classification as tree category determination, and maximum entropy and boosting model with subtree features were used. The experiment results showed that the subtree mining method can achieve a higher accuracy in question classification task.

In this paper, we formulate the *what-type* question classification as word sequence tagging problem. The tagged label is either one of the question categories for nouns s or "O" for other words. Since *head noun* can be the discriminator for a question, its tag is extracted as the question category in our work. A long-dependency Conditional Random Fields Classifier is defined to tag question words with the features which not only include the syntactic and semantic features, but also the semantic categories' transition features.

3 Conditional Random Fields

Conditional Random Fields (CRFs) are a type of discriminative probabilistic model proposed for labeling sequential data (Lafferty et al. 2001). Its definition is as follows:

Definition: Let $G = (V, E)$ *be a graph such that* $\mathbf{Y} = (Y_v)_{v \in V}$, *so that* $\mathbf{Y}$ *is indexed by the vertices of* G. *Then* $(\mathbf{X}, \mathbf{Y})$ *is a conditional random field in case, when conditioned on* $\mathbf{X}$, *the random variables* Y_v *obey the Markov property with respect to the graph:* $p(Y_v \mid \mathbf{X}, Y_w, w \neq v) = p(Y_v \mid \mathbf{X}, Y_w, w \sim v)$, *where* $w \sim v$ *means that* w *and* v *are neighbors in* G.

The joint distribution over the label sequence $\mathbf{Y}$ given $\mathbf{X}$ has the form

$$p(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \exp\left(\sum_{e \in E, i} \lambda_i t_i(e, \mathbf{Y} \mid e, \mathbf{X}) + \sum_{v \in V, i} \mu_i s_i(v, \mathbf{Y} \mid v, \mathbf{X}) \right),$$

where $Z(\mathbf{X})$ is a normalization factor, s_i is a *state* feature function and t_i is a *transition* feature function, λ_i and μ_i are the corresponding weights.

Here we assume the features are given, then the parameter estimation problem is to determine the parameters $\theta = (\lambda_1, \lambda_2, \ldots, \mu_1, \mu_2, \ldots)$ from training data. The inference problem is to find the most probable label sequence $\hat{\mathbf{y}}$ for input sequence $\mathbf{x}$.

In the training set, we label all the noun words with semantic question categories, and other words will be labeled by "O". We suppose that only adjacent noun words connect with each other, and there is no edge between noun and non-noun words, i.e., noun word and non-noun words may share neighbor's state features, but they are not connected by an edge. A labeled example is shown as "What/O was/O Queen/HUM:individual Victria/HUM:individual 's/O title/HUM:title regarding/O India/LOC:-country". In this labeled sentence, only three edges connect four noun words: Queen, Victria, title and India.

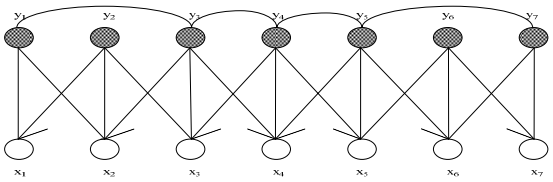

Figure 1. Long-Dependency CRFs, the dotted lines summarize many outgoing edges

With this assumption, we define a Long-Dependency Conditional Random Fields (LDCRFs) model (see Figure 1). The long dependency means that the target words may have no edge with its neighbors, but connect with other words at a long distance. It can be considered as a type of linear- chain CRFs. Its parameter estimation problem and inference problem can be solved by the algorithm for chain-structure CRFs (Sutton and McCallum, 2007).

4 Feature Sets

One of the most attractive advantages of CRFs is that they can integrate rich features, including not only state features, but also transition features. In this section, we will introduce the syntactic, semantic and transition features used in our sequence tagging approach.

4.1 Syntactic Features

The questions, which have similar syntactic style, intend to belong to the same category. Besides words, part-of-speech, chunker, parser information and question length are used as syntactic features.

All the words are lemmatized to root forms, and a window size (here is 4) is set to utilize the surrounding words.

The part-of-speech (POS) tagging is completed by SS Tagger (Tsuruoka and Tsujii, 2005), with our own improvement.

The noun phrase chunking (NP chunking) module uses the basic NP chunker software from

(Ramshaw and Marcus, 1995) to recognize the noun phrases in the question.

The importance of question syntactic structure is reported in (Zhang and Lee, 2002; Nguyen et al. 2007). They used complex machine learning method to capture the tree architecture. The LDCRFs based approach just selects parent node, relation with parent and governor for each target word generated from Minipar(Lin, 1999).

The length of question is another important syntactic feature. In our experiment, a threshold is set to denote the length as "high" or "low".

4.2 Semantic Features

Semantic features concern what words mean and how these meanings combine in sentence to form sentence meanings. Named Entity is a predefined semantic category for noun word. WordNet (Fellbaum, 1998) is a public semantic lexicon for English language, and it is used to get hypernym for noun word and synset for head verb which is the first notional verb in the sentence.

Named Entity: Named entity recognizer assigns a semantic category to the noun phrase. It is widely used to provide semantic information in text mining. In this paper, Stanford Named Entity Recognizer (Finkel et al. 2005) is used to classify noun phrases into four semantic categories: PERSON, LOCATION, ORGANIZARION and MISC.

Noun Hypernym: Hypernyms can be considered as semantic abstractions. It helps to narrow the gap between training set and testing set. For example, "What is Maryland's state bird?", if we recursively find the bird's hypernym "animal", which appeared in training set, this question can be easily classified.

In training set, we try to select appropriate hypernyms for each category. An correct WordNet sense is first assigned for each polysemous noun, and then all its hypernyms are recursively extracted. The sense determination step is processed with the algorithm in (Pedersen et al. 2005). They disambiguate word sense by assigning a target word the sense, which is most related to the senses of its neighboring words.

Since the word sense disambiguation method has low performance, with F1-measure below 50% reported in (Pedersen et al. 2005), a feature selection method is used to extract the most discriminative hypernyms. The hypernyms selection method is processed as follows: we first remove the low frequency hypernyms, and select the hypernyms using a chi-square method. The chi-square value measures the lack of independence between a hypernym *h* and category c_j. It is defined as:

$$\chi^2(h, c_j) = \frac{(A+B+C+D)\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)}$$

where *A* is the number of hypernym *h*, which belongs to category c_j; *B* is the number of *h* out of c_j; *C* is the number of other hypernyms in c_j; *D* is the number of other hypernyms out of c_j.

We set a threshold to select the most discriminative hypernym set. Extracted examples are shown in Figure 2.

ENTITY:animal:
animal, carnivore, chordate, equine, horse, living_thing, vertebrate, mammal, odd-toed_ungulate, organism, placental

ENTITY:food:
alcohol, beer, beverage, brew, cereal, condiment, crop, drink, drug_of_abuse, flavorer, food, foodstuff, helping, indefinite_quantity, ingredient, liquid, output, produce, small_indefinite_quantity, production, solid, substance, vegetable

Figure 2. Examples of extracted hypernym

It can be seen that these hypernyms are appropriate to describe the semantic meaning of the category. They are expected to work as the class-specific relational features which are semi-constructed by (Li and Roth, 2002). In our approach, we just use the noun's minimum upper hypernym, existing in training set, as the feature.

Head Verb Synset: To avoid losing question verb information, we extract head verb, which is the first notional verb in a question, and expand it using WordNet synset as feature. The head verb extraction is based on the following simple rules:

If the first word is "name" or "list", the head verb will be denoted as this word. If the first verb following question word is "do" or other auxiliary verb, the next verb is extracted as head verb. Otherwise the head verb is extracted as the first verb after question word.

4.3 Transition Features

State transition feature captures the contextual constraints among labels. We define it as

$$t_{y',y}(e=<i,i+1>, \mathbf{Y} \mid e, \mathbf{X}) = \delta(\mathbf{Y} \mid e=<y', y>).$$

Where *e* represents the edge between adjacent nouns. It captures adjacent categories as features to tag the target noun. Note that, for simplicity, the value of above feature is independent of the observations X.

5 Head Noun Extraction

After tagging all the words in a question, we will extract *head noun* and assign its tagged label to the question as the final question classification result.

The *head noun* extraction is a simple heuristic method inspired by (Metzler and Croft, 2005). We first run a POS tagger on each question, and post-process them to make sure that each sentence has at least one noun word. Next, the first NP chunk after the question word is extracted by shallow parsing. The *head noun* is determined by the following heuristic rules:

1. If the NP chunker is before the first verb, or the NP chunk is after the first verb but there is no possessive case after the NP chunker, we mark the rightmost word in the chunker as *head noun*.
2. Otherwise, extract the next NP chunker and recursively process the above rules.

Although this method may depend on the performance of POS tagger and shallow parser, it achieves the accuracy of over 95% on the UIUC data set in our implementation.

6 Experiments

6.1 Experiment Settings

Data Set:

We evaluate the proposed approach on the UIUC data set (Li and Roth, 2002). 5500 questions are selected for training, and 500 questions are selected for testing. The classification categories have been introduced as question ontology in section 1. This paper only focuses on 50 fine classes.

To train the LDCRFs, we manually labeled all the noun words with one of 50 fine categories. Other words are labeled with "O". One of the labeled examples is "What/O was/O Queen/HUM:individual Victria/HUM:individual 's/O title/HUM:title regarding/O India/LOC:country". Ten people labeled 3407 *what-type* questions as training set. Each question was independently annotated by two people and reviewed by the third. For words which have more than one category, the annotators selected the most salient one according to the context. For testing set, 351 *what-type* questions were selected for experiments evaluation.

Evaluation metric:

Accuracy performance is widely used to evaluate question classification methods [Li and Roth, 2002; Zhang and Lee, 2003, Melter and Croft, 2004; Nguyen et al. 2007].

6.2 Approach Performance Evaluation

	# Wrong	Accuracy
SVM	86	75.50%
LDCRFs-based	80	77.20%

Table 3. LDCRFS-based Approach V.S. SVM

Table 3 shows the compared results between the proposed LDCRFs based approach and SVM with unigram feature. The LDCRFs based approach achieves accuracy of 77.20%, compared with 75.50% of SVM. Observing the detailed classification results, we conclude two advantages of LDCRFs over SVMs. First LDCRFs based approach focuses on *head noun* to reduce the noise generated by other words. The question "What is the length of the coastline of the state of Alaska?" is misclassified as "LOC:state" by SVM, whereas it is correctly classified by our approach. Second, LDCRFs based approach can utilize rich features, including not only state features, but also transition features. With the new features involved, LDCRFs is expected to improve classification performance. This unigram result is used as our baseline. The following experiments are conducted to test the new feature contribution.

Syntactic Features:

In addition to words, four types of features, including part-of-speech (POS), chunker, parser information (Parser), and question length (Length), are extracted as syntactic features.

	Accuracy
Unigram (U)	77.20%
U+POS	78.35%
U+Chunker	77.20%
U+Parser	79.20%
U+Length	77.49%
Total Syn	80.06%

Table 4. Syntactic Feature Performance

From the syntactic feature results in Table 4, we can draw the following conclusions:

(a). Among four types of syntactic features, pars-

er information contributes mostly. (Metzler and Croft, 2005) once claimed that it didn't make improvement by just incorporating these information as explicit feature, and they should be used implicitly via a tree structure. Without using the complex tree mining and representing technique, our LDCRFs-based approach just incorporates the word parent, relation with parent and word governor from Minipar as features. The experiments show that the parser information feature is able to capture the syntactic structure information, and it makes much improvement in this sequence tagging approach.

(b) Question length makes small improvement. However, the chunker features make no improvement, consistent with the observation reported by (Li and Roth, 2006).

© The best accuracy (80.06%) is achieved by integrating all the syntactic features.

Semantic Features:

	Accuracy
Unigram(U)	77.20%
U+NE	77.20%
U+HVSyn	78.63%
U+NHype	78.35%
Total Sem	80.06%

Table 5. Semantic Feature Performance

The semantic features include Named Entity (NE), Noun Hypernym (NHype) and Head Verb Synset (HVSyn).

From Table 5 we can draw the following conclusions:

(a) NE makes no improvement in classification task. The reason is that the named entity recognizer contains only four semantic categories. It is too coarse to distinguish 50 fined-categories.

(b) The LDCRFs-based approach just considers the noun words as semantic words. The head verb synsets (HVSyn) are imported as one of semantic features. The experiment results show that it is effective to incorporate the head verb as features, which achieves the best individual accuracy among semantic features.

(c) Noun hypernyms (NHype) are the most important semantic features. They narrow the semantic gap between training set and testing set. From Section 4.2, we can see that the selected noun hypernyms are appropriate for each category. While, the experiment with NHype features doesn't make considerable improvement as we previously thought. The reason may come from the fact that the word sense disambiguation method has low performance. A hypernym selection method is used in training set, but we didn't tackle the error in testing set. Once the word sense disambiguation is wrong, it will not make improvement, but generate noise (see the discussion examples in next section).

(d) It is an interesting result that using all the semantic features can achieve the same accuracy as the syntactic features (80.06%).

Feature Combination:

In this section, we carry out experiments to examine whether the performance can be boosted by integrating syntactic features and semantic features. Several results are shown in Table 6. The experiments show that:

(a) Parser Information and Head Verb Synset are both the most contributive features for syntactic set and semantic feature set. While the performance with these two features can't beat the performance by combining Parser Information and Noun Hypernyms.

	Accuracy
U+POS+NE+HVSyn	80.91%
U+Parser+NHype	81.77%
U+Parser+HVSyn	80.91%
U+POS+Length+NHype	80.63%
Total	82.05%

Table 6. Combined Feature Performance

(b) The best result for classifying *what-type* questions with our approach is achieved by integrating all the features. The accuracy is 82.05%, which is 18.7 percent error reduction (from 22.08% to 17.95%) over unigram feature set. It shows that the features we extract are effectively used in our CRFs based approach.

Transition Feature:

Transition feature can capture the information between adjacent categories. It offers another semantic feature for LDCRFs-based approach.

	No transition features	With transition features
Syn	79.20%	80.06%
Sem	79.49%	80.06%
Total	81.48%	82.05%

Table 7. Transition Feature Performance

The performances of all these three experiment decline without the transition features. It shows that the dependency between adjacent se-

mantic categories contributes to the classifier performance.

6.3 System Performance Comparison and Discussion

In this section, the *what-type* questions and non-*what-type* questions are combined to show the final result. Non-*what-type* questions are classified using SVM with unigrams as reported in Section 1, and *what-type* questions are classified by the LDCRFs based approach. The combined results are used to compare with the current question classification methods.

Classifier	Accuracy
Li's Hierarchical method	84.20%
Nguyen's tree method	83.60%
Metzler's U+ WordNet method	82.20%
LDCRFs-based with U+Parser	83.60%
LDCRFs-based with U+NHype	83.00%
LDCRFs-based with total features	85.60%

Table 8. Comparison with related work

Table 8 shows the accuracies of the LDCRFs based question classification approach with different feature sets, in comparison with the tree method (Nguyen et al. 2007), the WordNet Method (Metzler and Croft, 2005) and the hierarchical method (Li and Roth, 2002). We can see the LDCRFs-based approach is effective:

(a) Without formulating the syntactic structure as a tree, the LDCRFs-based approach still achieves accuracy 83.60% with unigram and parser information, which is the same as Nguyen's tree classifier.

(b) Although the LDCRFs-based approach with unigrams and Noun Hypernyms generates noise as described in Section 6.2, it still outperforms the Metzler's method using WordNet and unigram features (83.00% v.s. 82.20%).

(c) The experiment with total features achieves the accuracy of 85.60%. It outperforms Li's Hierarchical classifier, even they use semi-automatic constructed features.

6.3.1 Analysis and Discussion

Even the sequence tagging model achieves high accuracy performance, there still exists many problems. We use the matrix defined in Li and Roth (2002) to show the performance errors. The metric is defined as follows:

$$D_{ij} = Err_{ij} * 2/(N_i + N_j)$$

Where Err_{ij} is the number of questions in class *i* that are misclassified as belong to class *j*, *Ni* and *Nj* are the numbers of questions in class *i* and *j* respectively.

From the matrix in Figure 3, we can see two major mistake pairs are "ENTY:substance" and "ENTY:other", "ENTY:currency" and "NUM:money". They really have similar meanings, which confuses even human beings.

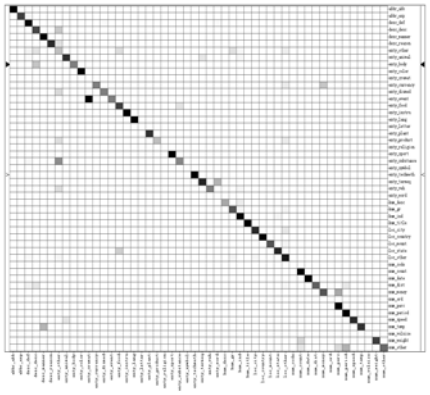

Figure 3. The gray-scale map of Matrix D[n,n]. The gray scale of the small box in position [i,j] denotes *D[i,j]*. The larger *Dij* is, the darker the color is.

Several factors influence the performance:

(a) *Head noun* extraction error: This error is mainly caused by errors of POS tagger and shallow parser. For the wrong POS example "what/WP hemisphere/EX is/VBZ the/DT Philippines/NNPS in/IN ?/.", "Philippines" is extracted as head word. The result is misclassified into "LOC:country". For the shallow parser error example "what/WP/B-NP is/VBZ/B-VP the/D T/BNP speed/NN/I-NP humminbirds/NNS /I-NP fly/V- BP/B-VP ?/./O", "hummingbirds" is extract as head word, rather than "speed". The question is misclassified into "ENTY:animal".

(b) WordNet sense disambiguation errors: In question "What is the highest dam in the U.S. ?" The real sense for dam is dam#1: a barrier constructed to contain the flow of water or to keep out the sea; while the disambiguation method determine the second sense as dam#2: a metric unit of length equal to ten meters.

(c) Lack of *head nouns*: the CRFs based approach is sensitive to the *Head Noun*. If the question doesn't contain the *head noun*, it is difficult to produce the correct result, such as the question "What is done with worn or outdated flags?" In the future work, we will focus on the *head noun* absence problem.

7 Conclusion

In this paper, we propose a novel approach with Conditional Random Fields to classify *what-type* questions. We first use the CRFs model to label all the words in a question, and then choose the label of *head noun* as the question category. As far as we know, this is the first trial to formulate question classification into word sequence tagging problem. We believe that the model has two distinguished advantages:

1. Extracting *head noun* can eliminate the noise generated by the non-head words
2. The Conditional Random Fields model can integrate rich features, including not only the syntactic and semantic features, but also the transition features between labels.

Experiments show that the LDCRFs-based approach can achieve comparable performance to those of the state-of-the-art question answering systems. With the addition of more features, the performance of the LDCRFs based approach can be expected to be further improved.

Acknowledgement

This work is supported by National Natural Science Foundation of China (60572084, 60621062), Hi-tech Research and Development Program of China (2006AA02Z321), National Basic Research Program of China (2007CB311003). Thank Shuang Lin and Jiao Li for revising this paper. Thanks for the reviewers' comments.

References

Christiane Fellbaum. 1998. *WordNet: an Electronic Lexical Database.* MIT Press.

Prager, J., D. Radev, E. Brown, A. Coden, and V. Samn. 1999. *`The use of predictive annotation for question answering in TREC'.* In: Proceedings of the 8th Text Retrieval Conference (TREC-8).

John Lafferty, Andrew McCallum, Fernando Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* In *Proceedings of* ICML-2001.

Li, X. and D. Roth. 2002. *Learning question classifiers.* In Proceedings of the 19th International Conference on Compuatational Linguistics (COLING), pages 556–562.

Zhang, D. and W. Lee. 2003. *Question classification using support vector machines*. In Proceedings of the 26th Annual International ACM SIGIR conference, pages 26–32

Donald Metzler and W. Bruce Croft. 2004. *Analysis of Statistical Question Classfication for Fact-based Questions.* In Journal of Information Retrieval.

Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan . 2005. *Maximizing Semantic Relatedness to Perform Word Sense Disambiguation.* University of Minnesota Supercomputing Institute Research Report UMSI 2005/25, March.

Xin Li, Dan Roth. 2006. *Learning Question Classifiers: The Role of Semantic Information.* In Natural Language Engineering, 12(3):229-249

Minh Le Nguyen, Thanh Tri Nguyen and Akira Shimazu. 2007. *Subtree Mining for Question Classification Problem.* In Proceedings of the 20th International Conference on Artificial Intelligence. Pages 1695-1700.

C. Sutton and A. McCallum. 2007. *An introduction to conditional random fields for relational learning*. In L. Getoor and B. Taskar (Eds.). Introduction to statistical relational learning. MIT Press.

Y. Tsuruoka and J. Tsujii,. 2005. *Bidirectional inference with the easiest-first strategy for tagging sequence data.* In Proc. HLT/EMNLP'05, Vancouver, October, pp. 467-474.

L. Ramshaw and M. Marcus. 1995. *Text chunking using transformation-based learning*, Proc. 3rd Workshop on Very Large Corpora, pp. 82–94.

J.R. Finkel, T. Grenager and C. Manning. 2005. *Incorporating non-local information into information extraction systems by Gibbs sampling*. Proc. 43rd Annual Meeting of ACL, pp. 363–370.

D. Lin. 1999. *MINIPAR: a minimalist parser. In Maryland Linguistics Colloquium*, University of Maryland, College Park.