Question Classification using Head Words and their Hypernyms

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

Zengchang Qin
EECS Department
University of California
at Berkeley
CA 94720-1776, USA
zqin@cs.berkeley.edu

Abstract

Question classification plays an important role in question answering. Features are the key to obtain an accurate question classifier. In contrast to Li and Roth (2002)'s approach which makes use of very rich feature space, we propose a compact yet effective feature set. In particular, we propose head word feature and present two approaches to 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, our linear SVM and Maximum Entropy (ME) models reach the accuracy of 89.2% and 89.0% respectively over a standard benchmark dataset, which outperform the best previously reported accuracy of 86.2%.

1 Introduction

An important step in question answering (QA) and other dialog systems 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. For instance, the classification of question *What is autism* to a definition type question would trigger the search strategy specific for definition type (e.g., using predefined templates like: *Autism is ...* or *Autism is defined as...*). In fact, the combination of QA 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. Considering also examples (Li and Roth, 2006) *What tourist attractions are there in Reims*, *What are the names of the tourist attractions in Reims*, *What do most tourists visit in Reims*, *What attracts tourists to Reims*, and *What is worth seeing in Reims*, all these reformulations are of the same answer type of location. Different wording and syntactic structures make it difficult for classification.

Many QA systems used manually constructed sets of rules to map a question to a type, which is not efficient in maintain and upgrading. With the increasing popularity of statistical approaches, machine learning plays a more and more important role in this task. A salient advantage of machine learning approach is that one can focus on designing insightful features, and rely on learning process to efficiently and effectively cope with the features. In addition, a learned classifier is more flexible to reconstruct than a manually constructed system because it can be trained on a new taxonomy in a very short time. Earlier question classification work includes Pinto et al. (2002) and Radev et at. (2002), in which language model and Rappier rule learning were employed respectively. More recently, Li and Roth (2002) have developed a machine learning approach which uses the SNoW learning architecture (Khardon et al.,

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 927–936,
Honolulu, October 2008.

1999). They have compiled the UIUC question classification dataset [1] which consists of 5500 training and 500 test questions. The questions in this dataset are collected from four sources: 4,500 English questions published by USC (Hovy et al., 2001), about 500 manually constructed questions for a few rare classes, 894 TREC 8 and TREC 9 questions, and also 500 questions from TREC 10 which serve as the test dataset. All questions in the dataset have been manually labeled by them according to the coarse and fine grained categories as shown in Table 3, with coarse classes (in bold) followed by their fine class refinements. In addition, the table shows the distribution of the 500 test questions over such categories. Li and Roth (2002) have made use of lexical words, part of speech tags, chunks (non-overlapping phrases), head chunks (the first noun chunk in a question) and named entities. They achieved 78.8% accuracy for 50 fine grained classes. With a hand-built dictionary of semantically related words, their system is able to reach 84.2%.

The UIUC dataset has laid a platform for the follow-up research. Hacioglu and Ward (2003) used linear support vector machines with question word bigrams and error-correcting output to obtain accuracy of 80.2% to 82.0%. Zhang and Lee (2003) used linear SVMs with all possible question word grams, and obtained accuracy of 79.2%. Later Li and Roth (2006) used more semantic information sources including named entities, WordNet senses, class-specific related words, and distributional similarity based categories in question classification task. With all these semantic features plus the syntactic ones, their model was trained on 21'500 questions and was able to achieve the best accuracy of 89.3% on a test set of 1000 questions (taken from TREC 10 and TREC 11) for 50 fine classes. Most recently, Krishnan et al. (2005) used a short (typically one to three words) subsequence of question tokens as features for question classification. Their model can reach the accuracy of 86.2% using UIUC dataset over fine grained question categories, which is the highest reported accuracy on UIUC dataset.

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. In particular, we propose head word feature and present two approaches to 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 89.2% using linear SVMs, or 89.0% using ME for 50 fine classes.

[1] available at http://12r.cs.uiuc.edu/∼cogcomp/Data/QA/QC

2 Classifiers

In this section, we briefly present two classifiers, support vector machines and maximum entropy model, which will be employed in our experiments. These two classifiers perform roughly identical in the question classification task.

2.1 Support Vector Machines

Support vector machine (Vapnik, 1995) is a useful technique for data classification. Given a training set of instance-labeled pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, l$ where $\mathbf{x}_i \in R^n$ and $\mathbf{y} \in \{1, -1\}^l$, the support vector machines (SVM) require the solution of the following optimization problem: $\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$ subject to $y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$. Here training vectors $\mathbf{x}_i$ are mapped into a higher (maybe infinite) dimensional space by the function ϕ. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_i)$ is called the kernel function. There are four basic kernels: linear, polynomial, radial basis function, and sigmoid. In the question classification context, $\mathbf{x}_i$ is represented by a set of binary features, for instance, the presence or absence of particular words. $y_i \in \{1, -1\}$ indicates wether a question is of a particular type or not. Due to the large number of features in question classification, one may not need to map data to a higher dimensional space. It has been commonly accepted that the linear kernel of $K(\mathbf{x}_i, \mathbf{x}_j) = {\mathbf{x}_i}^T\mathbf{x}_i$ is good enough for question classification. In this paper, we adopt the LIBSVM (Chang and Lin, 2001) implementation in our experiments.

2.2 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. In the context of question classification, a sample feature could be the presence of a particular word associated with a particular question type. The maximum entropy model is the model with maximum entropy of all models that satisfy the constraints. In this paper, we adopt Stanford Maximum Entropy (Manning and Klein, 2003) implementation in our experiments.

3 Features

Each question is represented as a bag of features and is feeded into classifiers in training stage. We present five binary feature sets, namely question wh-word, head word, WordNet semantic features for head word, word grams, and word shape feature. The five feature sets will be separately used by the classifiers to determine their individual contribution. In addition, these features are used in an incremental fashion in our experiments.

3.1 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*. We have adopted eight question wh-words, namely *what*, *which*, *when*, *where*, *who*, *how*, *why*, and *rest*, with *rest* being the type does not belong to any of the previous type. For example, the question *Name a food high in zinc* is a *rest* type question.

3.2 Head Word

Li and Roth (2002;2006) used head chunks as features. The first noun chunk and the first verb chunk after the question word in a sentence are defined as head chunks in their approach. Krishnan et al. (2005) used one contiguous span of tokens which is denoted as the *informer span* as features. In both approaches, noisy information could be introduced. For example, considering the question of *What is a group of turkeys called*, both the head chunk and informer span of this question is *group of turkeys*. The word of *turkeys* in the chunk (or span) contributes to the classification of type ENTY:animal if the hypernyms of WordNet are employed (as described in next section). However, the extra word *group* would introduce ambiguity to misclassify such question into HUMAN:group, as all words appearing in chunk are treated equally. To tackle this problem, we propose the feature of *head word*, which is one single word specifying the object that the question seeks. In the previous example *What is a group of turkeys called*, the head word is exactly *turkeys*. In doing so, no misleading word *group* is augmented. Another example is *George Bush purchased a small interest in which baseball team*. The head chunk, informer span and head word are *baseball team*, *baseball team* and *team* respectively. The extra word *baseball* in the head chunk and informer span may lead the question misclassified as ENTY:sport rather than HUM:group. In most cases, the head chunk or informer span include head words. The head chunk feature or informer span feature would be beneficiary so long as the useful information plays a stronger role than the misleading one. Nevertheless, this is not as effective as the introduction of one head word.

To obtain the head word feature, a syntactic parser is required. A syntactic parser is a model that outputs the grammatical structure of given sentences. There are accurate parsers available such as Chaniak parser (Charniak and Johnson, 2005), Stanford parser (Klein and Manning, 2003) and Berkeley parser (Petrov and Klein, 2007), among which we use the Berkeley parser [2] to help identify the head word. Figure 1 shows two example parse trees for questions *What year did the Titanic sink* and *What is the sales tax in Minnesota* respectively.

Collins rules (Collins, 1999) can be applied to parse trees to extract the syntactic head words. For example, the WHNP phrase (Wh-noun phrase) in the top of Figure 1 takes its WP child as its head word, thus assigning the word *what* (in the bracket) which is associated with WP tag to the syntactic head word of WHNP phrase. Such head word assignment is carried out from the bottom up and the word *did* is extracted as the head word of the whole question. Similarly, the word *is* is extracted as the

[2] available at http://nlp.cs.berkeley.edu/Main.html#parsing

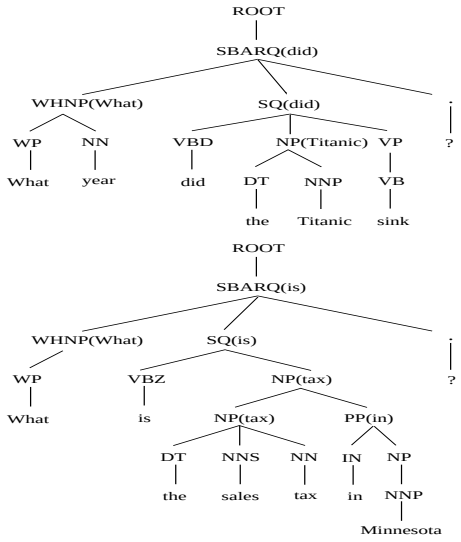

Figure 1: Two example parse trees and their head words assignment

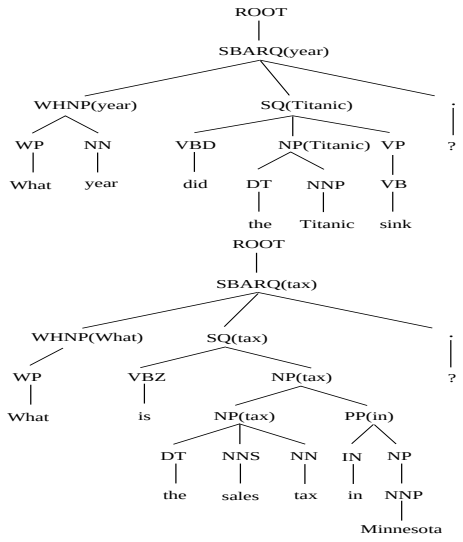

Figure 2: Two example parse trees and their revised head words assignment

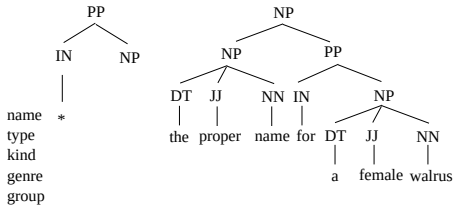

Figure 3: Post fix for the head word assignment

syntactic head word in the bottom of Figure 1.

Collins head words finder rules have been modified to extract semantic head word (Klein and Manning, 2003). To better cover the question sentences, we further re-define the semantic head finder rules to fit our needs. In particular, the rules to find the semantic head word of phrases SBARQ (Clause introduced by subordinating conjunction), SQ (subconstituent of SBARQ excluding wh-word or wh-phrase), VP (verb phrase) and SINV (declarative sentence with subject-aux inversion) are redefined, with the head preference of noun or noun phrase rather than verb or verb phrase. The new head word assignments for the previous two examples are shown in Figure 2.

If the head word is any of *name*, *type* or *kind* etc, post fix is required to identify the real head word if necessary. In particular, we compile a tree pattern as shown in the left of Figure 3. If this pattern is matched against a given parse question parse tree, the head word is re-assigned to the head word of NP node in the tree pattern. For example, the initial head word extracted from parse tree of question *What is the proper name for a female walrus* is *name*. As such parse tree (as shown partially in the right of Figure 3) matches the compiled tree pattern, the post operation shall fix it to *walrus*, which is the head word of the NP in the tree pattern. This post fix helps classify the question to ENTY:animal.

In addition to the question head word as described above, we introduce a few regular expression patterns to help question head word identification. Note that these patterns depend on the question type taxonomy as shown in Table 3. For example, considering the questions of *What is an atom* and *What are invertebrates*, the head word of *atom* and *invertebrates* do not help classify such questions to DESC:def. To resolve this, we create a binary feature using a string regular expression which begins with *what is/are* and follows by an optional *a*, *an*, or *the* and then follows by one or two words. If a question matches this regular expression, a binary feature (a placeholder word is used in implementation, for instance DESC:def_1 in this case) would be inserted to the feature set of the question. This feature, if it is beneficial, would be picked up by the classifiers (SVMs or MEs) in training. We list all regular expression patterns which are used in our experiments as following:

DESC:def pattern 1 The question begins with *what is/are* and follows

by an optional *a*, *an*, or *the* and then follows by one or two words.

DESC:def pattern 2 The question begins with *what do/does* and ends with *mean*.

ENTY:substance pattern The question begins with *what is/are* and ends with *composed of/made of/made out of*.

DESC:desc pattern The question begins with *what does* and ends with *do*.

ENTY:term The question begins with *what do you call*.

DESC:reason pattern 1 The question begins with *what causes/cause*.

DESC:reason pattern 2 The question begins with *What is/are* and ends with *used for*.

ABBR:exp pattern The question begins with *What does/do* and ends with *stand for*.

HUM:desc pattern The question begins with *Who is/was* and follows by a word starting with a capital letter.

It is worth noting that all these patterns serve as feature generators for given questions: the feature becomes active if the pattern matches the questions. The algorithm to extract question head word is shown in Algorithm 1. There is no head word returned for *when*, *where* or *why* type questions, as these hw-words are informative enough; the inclusion of other words would introduce noisy information. If the question is of type *how*, the word following *how* is returned as head word. The patterns are then attempted to match the question if it is of type *what* or *who*. If there is a match, the placehold word for such pattern (e.g., *HUM:desc* for HUM:desc pattern) is returned as head word. If none of the above condition is met, the candidate head word is extracted from the question parse tree using the redefined head finder rules. Such extracted head word is returned only if it has noun or noun phrase tag; otherwise the first word which has noun or noun phrase tag is returned. The last step is a back up plan in case none of the previous procedure happens.

Algorithm 1 Question head word extraction

Require: Question q
Ensure: Question head word
if q.type == *when*|*where*|*why* **then**
  return null
end if
if q.type == *how* **then**
  return the word following word "how"
end if
if q.type == *what* **then**
  for any aforementioned regular expression r (except HUM:desc pattern) **do**
    if(q matches r)
      return r.placehold-word
  end for
end if
if q.type == *who* && q matches HUM:desc pattern **then**
  return "HUM:desc"
end if
String $candidate$ = head word extracted from question parse tree
if $candidate$.tag starts with NN **then**
  return $candidate$
end if
return the first word whose tag starts with NN

3.3 WordNet Semantic Feature

WordNet (Fellbaum, 1998) is a large English lexicon in which meaningfully related words are connected via cognitive synonyms (synsets). The WordNet is a useful tool for word semantics analysis and has been widely used in question classification (Krishnan et al., 2005; Schlaefer et al., 2007). A natural way to use WordNet is via hypernyms: Y is a hypernym of X if every X is a (kind of) Y. For example, the question of *What breed of hunting dog did the Beverly Hillbillies own* requires the knowledge of *animal* being the hypernym of *dog*. In this paper, we propose two approaches to augment WordNet semantic features, with the first augmenting the hypernyms of head words as extracted in previous section directly, and the second making use of a WordNet similarity package (Seco et al., 2004), which implicitly employs the structure of hypernyms.

3.3.1 Direct Use of Hypernyms

In WordNet, senses are organized into hierarchies with hypernym relationships, which provides a natural way to augment hypernyms features from the original head word. For example, the hierarchies for a noun sense of domestic dog is described as: *dog* → *domestic animal* → *animal*, while another noun sense (a dull unattractive unpleasant girl or woman) is organized as *dog* → *unpleasant woman* → *unpleasant person*. In addition, a verb sense of dog is organized as *dog* → *pursue* → *travel*. In our first approach, we attempt to directly introduce hypernyms for the extracted head words. The augment of hypernyms for given head word can introduce useful information, but can also bring noise if the head word or the sense of head word are not correctly identified. To resolve this, three questions shall be addressed: 1) which part of speech senses should be augmented? 2) which sense of the given word is needed to be augmented? and 3) how many depth

are required to tradeoff the generality (thus more informative) and the specificity (thus less noisy). The first question can be answered by mapping the Penn Treebank part of speech tag of the given head word to its WordNet part of speech tag, which is one of POS.NOUN, and POS.ADJECTIVE, POS.ADVERB and POS.VERB. The second question is actually a word sense disambiguation (WSD) problem. The Lesk algorithm (Lesk, 1986) is a classical algorithm for WSD. It is based on the assumption that words in a given context will tend to share a common topic. A basic implementation of the The Lesk algorithm is described as following:

1. Choosing pairs of ambiguous words within a context
2. Checks their definitions in a dictionary
3. Choose the senses as to maximize the number of common terms in the definitions of the chosen words

In our head word sense disambiguation, the context words are words (except the head word itself) in the question, and the dictionary is the gloss of a sense for a given word. Algorithm 2 shows the adapted Lesk algorithm which is employed in our system. Basically, for each sense of given head word, this algorithm computes the maximum number of common words between gloss of this sense and gloss of any sense of the context words. Among all head word senses, the sense which results in the maximum common words is chosen as the optimal sense to augment hypernyms later. Finally the third question is answered via trail and error based on evaluating randomly generated 10% data from the training dataset. Generally speaking, if the identification of the head word is not accurate, it would brings significant noisy information. Our experiments show that the use of depth six produces the best results over the validation dataset. This indirectly proves that our head word feature is very accurate: the hypernyms introduction within six depths would otherwise pollute the feature space.

Algorithm 2 Head word sense disambiguation

Require: Question q and its head word h
Ensure: Disambiguated sense for h
int $count$ = 0
int $maxCount$ = -1
sense $optimum$ = null
for each sense s for h **do**
  $count$ = 0
  for each context word w in q **do**
    int $subMax$ = maximum number of common words in s definition (gloss) and definition of any sense of w
    count = count + sumMax
  end for
  if $count > maxCount$ **then**
    $maxCount = count$
    $optimum = s$
  end if
end for
return $optimum$

3.3.2 Indirect Use of Hypernyms

In this approach, we make use of the WordNet Similarity package (Seco et al., 2004), which implicitly employs WordNet hypernyms. In particular, for a given pair of words, the WordNet similarity package models the length of path traveling from one word to the other over the WordNet network. It then computes the semantic similarity based on the path. For example, the similarity between *car* and *automobile* is 1.0, while the similarity between *film* and *audience* is 0.38. For each question, we use the WordNet similarity package to compute the similarity between the head word of such question and each description word in a question categorization. The description words for a question category are a few words (usually one to three) which explain the semantic meaning of such a question category [3]. For example, the descriptions words for category ENTY:dismed are *diseases* and *medicine*. The question category which has the highest similarity to the head word is marked as a feature. This is equal to a mini question classifier. For example, as the head word *walrus* of question *What is the proper name for a female walrus* has the highest similarity measure to *animals*, which is a description word of category ENTY:animal, thus the ENTY:animal is inserted into the feature set of the given question.

3.4 N-Grams

An N-gram is a sub-sequence of N words from a given question. Unigram forms the bag of words feature, and bigram forms the pairs of words feature, and so forth. We have considered unigram, bigram, and trigram features in our experiments. The reason to use such features is to provide word sense

[3] available at http://12r.cs.uiuc.edu/∼cogcomp/Data/QA/QC/definition.html

disambiguation for questions such as *How long did Rip Van Winkle sleep*, as *How long* (captured by wh-word and head word features) could refer to either NUM:dist or NUM:period. The word feature of *sleep* help determine the NUM:period classification.

3.5 Word Shape

Word shape in a given question may be useful for question classification. For instance, the question *Who is Duke Ellington* has a mixed shape (begins a with capital letter and follows by lower case letters) for *Duke*, which roughly serves as a named entity recognizer. We use five word shape features, namely all upper case, all lower case, mixed case, all digits, and other. The experiments show that this feature slightly boosts the accuracy.

4 Experimental Results

We designed two experiments to test the accuracy of our classifiers. The first experiment evaluates the individual contribution of different feature types to question classification accuracy. In particular, the SVM and ME are trained from the UIUC 5500 training data using the following feature sets: 1) wh-word + head word, 2) wh-word + head word + direct hypernym, 3) wh-wod + head word + indirect hypernym, 4) unigram, 5) bigram, 6) trigram, and 7) word shape. We set up the tests of 1), 2) and 3) due to the fact that wh-word and head word can be treated as a unit, and hypernym depends on head word. In the second experiment, feature sets are incrementally feeded to the SVM and ME. The parameters for both SVM and ME classifiers (e.g., the C in the SVM) are all with the default values. In order to facilitate the comparison with previously reported results, the question classification performance is measured by accuracy, i.e., the proportion of the correctly classified questions among all test questions.

4.1 Individual Feature Contribution

Table 1 shows the question classification accuracy of SVM and ME using individual feature sets for 6 coarse and 50 fine classes. Among all feature sets, wh-word + head word proves to be very informative for question classification. Our first WordNet semantic feature augment, the inclusion of direct hypernym, can further boost the accuracy in the fine classes for both SVM and ME, up to four percent. This phenomena conforms to Krishnan et al. (2005) that WordNet hypernym benefits mainly on the 50 fine classes classification. Li and Roth (2006) made use of semantic features including named entities, WordNet senses, class-specific related words, and distributional similarity based categories. Their system managed to improve around 4 percent with the help of those semantic features. They reported that WordNet didn't contribute much to the system, while our results show that the WordNet significantly boosts the accuracy. The reason may be that their system expanded the hypernyms for each word in the question, while ours only expanded the head word. In doing so, the augmentation does not introduce much noisy information. Notice that the inclusion of various depth of hypernyms results in different accuracy. The depth of six brings the highest accuracy of 85.4% and 85.6% for SVM and ME under 50 classes, which is very competitive to the previously reported best accuracy of 86.2% (Krishnan et al., 2005).

Table 1: Question classification accuracy of SVM and ME using individual feature sets for 6 and 50 classes over UIUC dataset

		6 class		50 class	
		SVM	ME	SVM	ME
wh-word + head word		92.0	92.2	81.4	82.0
wh-word +	depth=1	92.0	91.8	84.6	84.8
head word +	depth = 3	92.0	92.2	85.4	85.4
direct hypernym	depth = 6	92.6	91.8	85.4	85.6
wh-word + head + indirect hypernym		91.8	92.0	83.2	83.6
unigram		88.0	86.6	80.4	78.8
bigram		85.6	86.4	73.8	75.2
trigram		68.0	57.4	39.0	44.2
word shape		18.8	18.8	10.4	10.4

Our second proposed WordNet semantic feature, the indirect use of hypernym, does not perform as good as the first approach; it only contributes the accuracy gain of 1.8 and 1.6 in the fine classes for SVM and ME respectively. The reason may be two fold: 1) the description words (usually one to three words) of question categories are not representative enough, and 2) the indirect use of hypernyms via the WordNet similarity package is not as efficient as direct use of hypernyms.

Among the surface words features, unigram feature perform the best with accuracy of 80.4% for SVM under 50 classes, and 88.0% for SVM under 6 classes. It is not surprising that the word shape

feature only achieves small gain in question classification, as the use of five shape type does not provide enough information for question classification. However, this feature is treated as an auxiliary one to boost a good classifier, as we will see in the second experiment.

4.2 Incremental Feature Contribution

Based on the individual feature contribution, we then trained the SVMs and MEs using wh-word, head word, direct hypernyms (with depth 6) of head word, unigram, and word shape incrementally. Table 2 shows the question classification accuracy (broken down by question types) of SVM and ME for 6 coarse and 50 fine classes. As can be seen, the main difficulty for question classification lies in the *what* type questions. SVM and ME perform roughly identical if they use the same features. For both SVM and ME, the baseline using the wh-head word and head word results in 81.4% and 82.0% respectively for 50 fine class classification (92.0% and 92.2% for 6 coarse classes). The incremental use of hypernym feature within 6 depths boost about four percent for both SVM and ME under 50 classes, while slight gain or slight loss for SVM and ME for 6 coarse classes. The further use of unigram feature leads to another three percent gain for both SVM and ME in 50 classes. Finally, the use of word shape leads to another 0.6% accuracy increase for both SVM and ME in 50 classes. The best accuracy achieved for 50 classes is 89.2% for SVM and 89.0% for ME. For 6 coarse classes, SVM and ME achieve the best accuracy of 93.4% and 93.6% respectively.

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.

Note that if we replace the bigram with unigram, SVM and ME achieve the overall accuracy of 88.4% and 88.0% respectively for 50 fine classes, and the use of trigram leads SVM and ME to 86.6% and 86.8% respectively. The inclusion of unigram, bigram and trigram together won't boost the accuracy, which reflects the fact that the bigram and trigram features cannot bring more information given that unigram, wh-word and head word features are present. This is because the useful information which are supposed to be captured by bigram or trigram are effectively captured by wh-word and head word features. The unigram feature thus outperforms bigram and trigram due to the fact that it is less sparse. In addition, if we replace the indirect use of hypernym with the direct use of hypernym, the overall accuracy is 84.6% and 84.8% for SVM and ME respectively. All these experiments conform to the individual features contributions as shown in Table 1.

For a better understanding of the error distribution with respect to the 50 question categories, Table 3 shows the precision and recall for each question type in the best result (89.2%) using SVM. It is not surprising that some of the categories are more difficult to predict such as ENTY:other and ENTY:product, while others are much easier such as HUMAN:individual, since the former are more semantically ambiguous than the latter.

Table 3: Precision and recall for fine grained question categories

Class	#	P	R	Class	#	P	R
ABBR	9			desc	7	75.0	85.7
abb	1	100	100	manner	2	100	100
exp	8	88.9	100	reason	6	85.7	100
ENTITY	94			**HUMAN**	65		
animal	16	94.1	100	group	6	71.4	83.3
body	2	100	50.0	individual	55	94.8	100
color	10	100	100	title	1	0.0	0.0
creative	0	100	100	desc	3	100	100
currency	6	100	100	**LOC**	81		
dis.med.	2	40.0	100	city	18	100	77.8
event	2	100	50.0	country	3	100	100
food	4	100	50.0	mountain	3	100	66.7
instrument	1	100	100	other	50	83.9	94.0
lang	2	100	100	state	7	85.7	85.7
letter	0	100	100	**NUM**	113		
other	12	45.5	41.7	code	0	100	100
plant	5	100	100	count	9	81.8	100
product	4	100	25.0	date	47	95.9	100
religion	0	100	100	distance	16	100	62.5
sport	1	100	100	money	3	100	33.3
substance	15	88.9	53.3	order	0	100	100
symbol	0	100	100	other	12	85.7	50.0
technique	1	100	100	period	8	72.7	100
term	7	100	85.7	percent	3	75.0	100
vehicle	4	100	75.0	speed	6	100	83.3
word	0	100	100	temp	5	100	60.0
DESC	138			size	0	100	100
definition	123	89.0	98.4	weight	4	100	75.0

Table 4 shows the summary of the classification accuracy of all models which were applied to UIUC dataset. 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. Our results are summarized in the last two rows.

Our classifiers are able to classify some chal-

Table 2: Question classification accuracy of SVM and ME using incremental feature sets for 6 and 50 classes

6 coarse classes									
Type	#Quest	wh+headword		+headword hypernym		+unigram		+word shape	
		SVM	ME	SVM	ME	SVM	ME	SVM	ME
what	349	88.8	89.1	89.7	88.5	89.7	90.3	90.5	91.1
which	11	90.9	90.9	100	100	100	100	100	100
when	26	100	100	100	100	100	100	100	100
where	27	100	100	100	100	100	100	100	100
who	47	100	100	100	100	100	100	100	100
how	34	100	100	100	100	100	100	100	100
why	4	100	100	100	100	100	100	100	100
rest	2	100	100	50.0	50.0	100	50.0	100	50.0
total	500	92.0	92.2	92.6	91.8	92.8	93.0	93.4	**93.6**
50 fine classes									
Type	#Quest	wh+headword		+headword hypernym		+unigram		+word shape	
		SVM	ME	SVM	ME	SVM	ME	SVM	ME
what	349	77.4	77.9	82.8	82.5	85.4	85.1	86.2	86.0
which	11	81.8	90.9	81.8	90.9	90.9	100	90.9	100
when	26	100	100	100	100	100	100	100	100
where	27	92.6	92.6	92.6	92.6	92.6	92.6	92.6	92.6
who	47	100	100	100	100	100	100	100	100
how	34	76.5	76.5	76.5	79.4	97.1	91.2	97.1	91.2
why	4	100	100	100	100	100	100	100	100
rest	2	0.0	0.0	50.0	50.0	0.0	50.0	0.0	50.0
total	500	81.4	82.0	85.4	85.6	88.6	88.4	**89.2**	89.0

lenge questions. For instance, the question *What is the proper name for a female walrus* has been correctly classified as ENTY:animal. However, it still has nearly ten percent error rate for 50 fine classes. The reason is three fold: 1) there are inherently ambiguity in classifying a question. For instance, the question *What is mad cow disease*, it could be either of type DESC:def or ENTY:dismed; 2) there are inconsistent labeling in the training data and test data. For instance, *What is the population of Kansas* is labeled with the type NUM:other while *What is the population of Arcadia , Florida* is labeled with type NUM:count. Another example, *What county is Chicago in* is labeled with type LOC:other while *What county is Phoenix , AZ in* is labeled with type LOC:city; and 3) The parser can produce incorrect parse tree which would result in wrong head word extraction. For instance, the head word extracted from *What is the speed hummingbirds fly* is *hummingbirds* (the correct one should be *speed*), thus leading to the incorrect classification of ENTY:animal (rather than the correct NUM:speed).

5 Conclusion

In contrast to Li and Roth (2006)'s approach which makes use of very rich feature space, we proposed a compact yet effective feature set. In particular, we proposed head word feature and presented two approaches to augment semantic features of such head words using WordNet. In addition, Lesk's word sense disambiguation algorithm was adapted and the depth of hypernym feature was optimized through cross validation, which was to introduce useful information while not bringing too much noise. With further augment of wh-word, unigram feature, and word shape feature, we can obtain accuracy of 89.2% using linear SVMs, or 89.0% using ME for 50 fine classes.

Table 4: Classification accuracy of all models 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
Linear SVM	93.4	89.2
Maximum Entropy Model	93.6	89.0

6 Acknowledgments

This research was supported by British Telecom grant CT1080028046 and BISC Program of UC Berkeley.

References

Berger, A. L., V. J. D. Pietra, and S. A. D. Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, 22(1):39–71.

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

Charniak, E. and M. Johnson. 2005. Coarse-to-Fine N-Best Parsing and MaxEnt Discriminative Reranking. *The 43rd Annual Meeting on Association for Computational Linguistics*.

Collins, M. 1999. Head-Driven Statistical Models for Natural Language Parsing. *PhD thesis*, University of Pennsylvania.

Fellbaum, C. 1998. An Electronic Lexical Database. The MIT press.

Hacioglu, K. and W. Ward. 2003. Question Classification with Support Vector Machines and Error Correcting Codes. *The Association for Computational Linguistics on Human Language Technology*, vol. 2, pp. 28–30.

Hovy, E., L. Gerber, U. Hermjakob, C. Y. Lin, and D. Ravichandran. 2001. Towards Semantics-based Answer Pinpointing. *The conference on Human language technology research (HLT)*, pp. 1–7.

Khardon, R., D. Roth, and L. G. Valiant. 1999. Relational Learning for NLP using Linear Threshold Elements. *The Conference on Artificial Intelligence*, pp. 911–919.

Klein, D. and C. Manning. 2003. Accurate Unlexicalized Parsing. *The Association for Computational Linguistics*, vol. 1, pp. 423–430.

Krishnan, V., S. Das, and S. Chakrabarti. 2005. Enhanced Answer Type Inference from Questions using Sequential Models. *The conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Lesk, Michael. 1986. Automatic Sense Disambiguation using Machine Readable Dictionaries: how to tell a pine cone from an ice cream cone. *ACM Special Interest Group for Design of Communication Proceedings of the 5th annual international conference on Systems documentation*, pp. 24–26.

Li, X. and D. Roth. 2002. Learning Question Classifiers. *The 19th international conference on Computational linguistics*, vol. 1, pp. 1–7.

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

Manning, C. and D. Klein. 2003. Optimization, Maxent Models, and Conditional Estimation without Magic. *Tutorial at HLT-NAACL 2003 and ACL 2003*.

Petrov, S. and D. Klein. 2007. Improved Inference for Unlexicalized Parsing. *HLT-NAACL*.

Pinto, D., M. Branstein, R. Coleman, W. B. Croft, M. King, W. Li, and X. Wei. 2002. QuASM: A System for Question Answering using semi-structured Data *The 2nd ACM/IEEE-CS joint conference on Digital libraries.*.

Radev, D., W. Fan, H. Qi, H. Wu, and A. Grewal. 2002. Probabilistic question answering on the web. *The 11th international conference on World Wide Web*.

Schlaefer, N., J. Ko, J. Betteridge, S. Guido, M. Pathak, and E. Nyberg. 2007. Semantic Extensions of the Ephyra QA System for TREC 2007. *The Sixteenth Text REtrieval Conference (TREC)*.

Seco, N., T. Veale, and J. Hayes. 2004. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. *Proceedings of the European Conference of Artificial Intelligence*.

Vapnik, V. N. 1995. The Nature of Statistical Learning Theory. Springer-Verlag New York.

Voorhees, E. M. and H. T. Dang. 2005. Overview of the TREC 2005 Question Answering Track. *The Text Retrieval Conference (TREC2005)*.

Zhang D. and W. S. Lee. 2003. Question Classification using Support Vector Machines. *The ACM SIGIR conference in informaion retrieval*, pp. 26–32.