Learning Question Classifiers: The Role of Semantic Information†‡

Xin Li and Dan Roth

Department of Computer Science
University of Illinois at Urbana-Champaign
{xli1,danr}@uiuc.edu

(*Received 15 June 2004*)

Abstract

In order to respond correctly to a free form factual question given a large collection of text data, one needs to understand the question to a level that allows determining some of the constraints the question imposes on a possible answer. These constraints may include a semantic classification of the sought after answer and may even suggest using different strategies when looking for and verifying a candidate answer. This work presents the first work on a machine learning approach to question classification. Guided by a layered semantic hierarchy of answer types, we develop a hierarchical classifier that classifies questions into fine-grained classes.

This work also performs a systematic study of the use of semantic information sources in natural language classification tasks. It is shown that, in the context of question classification, augmenting the input of the classifier with appropriate semantic category information results in significant improvements to classification accuracy. We show accurate results on a large collection of free-form questions used in TREC 10 and 11.

1 Introduction

Open-domain question answering (Lehnert 1986; Harabagiu et al. 2000) and story comprehension (Hirschman et al. 1999) have become important directions in natural language processing. The purpose of the question answering (QA) task is to seek an *accurate and concise answer* to a free-form factual question[1] from a large collection of text data, rather than a full document, judged relevant as in standard information retrieval tasks. The difficulty of pinpointing and verifying the precise answer makes question answering more challenging than the common information

† This paper combines and extends early works in (Li and Roth 2002; Li, Small, and Roth 2004).

‡ Research supported by NSF grants IIS-9801638 and ITR IIS-0085836 and an ONR MURI Award.

[1] It does not address questions like 'Do you have a light?', which calls for an action, but rather only 'What', 'Which', 'Who', 'When', 'Where', 'Why' and 'How'-questions that ask for a simple fact.

retrieval task done by search engines. This difficulty is more acute in tasks such as story comprehension in which the textual resources are more confined and the target text is less likely to exactly match text in the questions. For this reason, advanced natural language techniques rather than key term extraction and expansion are needed.

Recent works (Hovy et al. 2001; Moldovan et al. 2002; Roth et al. 2002) have shown that locating an accurate answer hinges on first filtering out a wide range of candidates based on some categorization of answer types given a question. Specifically, this classification task has two purposes. First, it provides constraints on the answer types that allow further processing to precisely locate and verify the answer. For example, when considering the question

Q: *What Canadian city has the largest population?*,

we do not want to test every noun phrase in a document to see whether it provides an answer. The hope is, at the very least, to classify this question as having answer type **city**, implying that only candidate answers that are cities need consideration.

Second, it provides information that downstream processes may use in determining answer selection strategies that may be answer type specific. Besides the former example, the following examples also exhibit several aspects of this point:

Q: *What is a prism?*

Identifying that the target of this question is a **definition**, strategies that are specific for *definitions* (e.g., using predefined templates like: *A/The prism is...* or *Prisms are...*) may be useful. Similarly, in:

Q: *Why is the sun yellow?*

Identifying that this question asks for a **reason**, may lead to using a specific strategy for *reasons*.

Moreover, question classification would benefit question answering process further if it has the capacity to distinguish between a large and complex set of finer classes. A question classifier must take all of these into account and produce predictions appropriate for the downstream needs.

One way to exhibit the difficulty in manually building a question classifier (Harabagiu et al. 2000; Hermjakob 2001; Hovy et al. 2001) is to consider the reformulations of a single query:

- *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?*
- *What is worth seeing in Reims?*

All these reformulations target at the same answer type **Location**. However, different words and syntactic structures make it difficult for a manual classifier based on a small set of rules to generalize well and map all of these to the same answer type. This remains true even if external knowledge bases (e.g. WordNet (Fellbaum 1998)) are exploited to partially automate the mapping from word-level information to question classes (Harabagiu et al. 2000; Hermjakob 2001). State-of-the-art learning methods with appropriate features, on the other hand, may utilize the large number of potential features (derived from syntactic structures and lexical semantic analysis) to generalize and classify these cases automatically. By combining features from the same context, learning is also more robust to word sense ambiguity problem that might occur in the mapping.

This work develops a machine learning approach to question classification (QC) (Li and Roth 2002; Li, Small, and Roth 2004). The goal is to categorize questions into different semantic classes based on the possible semantic types of the answers. We develop a hierarchical classifier guided by a layered semantic hierarchy of answer types that makes use of a sequential model for multi-class classification (Even-Zohar and Roth 2001) and the SNoW learning architecture (Carlson et al. 1999). We suggest that it is useful to consider this classification task as a multi-class classification and find that it is possible to achieve good classification results despite the fact that the number of different labels used is fairly large, 50 fine-grained semantic classes.

At a high level, question classification may be viewed as a text categorization task (Sebastiani 2002). However, some characteristics of question classification make it different from the common task. On one hand, questions are relatively short and contain less word-based information compared with classifying the whole text. On the other hand, short questions are amenable for more accurate and deeper-level analysis. Our approach is, therefore, to augment the questions with syntactic and semantic analysis, as well as external semantic knowledge, as input to the text classifier.

In this way, this work on question classification can be also viewed as a case study in applying semantic information in text classification. This work systematically studies several possible semantic information sources and their contribution to classification. We compare four types of semantic information sources that differ in their granularity, the way they are acquired and their size: (1) automatically acquired named entity categories, (2) word senses in WordNet 1.7 (Fellbaum 1998), (3) manually constructed word lists related to specific categories of interest, and (4) automatically generated semantically similar word lists (Pantel and Lin 2002).

Our experimental study focuses on (1) testing the performance of the classifier in classifying questions into coarse and fine classes, and (2) comparing the contribution of different syntactic and semantic features to the classification quality. In the experiments, we observe that classification accuracies over $1,000$ TREC (Voorhees 2002) questions reach 92.5 percent for 6 coarse classes and 89.3 percent for 50 fine-grained classes, state-of-the-art performance for this task. We also observe that question classification is a indeed feature-dependent task and that semantic information is essential in order to achieve this level of accuracy. An error reduction of 28.7 percent can be achieved when semantic features are incorporated into the fine-grained classification.

The paper is organized as follows: Sec. 2 presents the question classification problem; Sec. 3 discusses the learning issues involved in QC and presents our learning approach; Sec. 4 explains how the sources of semantic information are incorporated as features and describes all the features defined in this task. Sec. 5 presents our experimental study and results. Related work is summarized in Sec. 6. In Sec. 7 we conclude by discussing a few issues left open by our study.

2 Question Classification

Many important natural language inferences can be viewed as problems of resolving ambiguity, either syntactic or semantic, based on properties of the surrounding context. These are typically modeled as classification tasks (Roth 1998). Examples include part-of-speech tagging where a word is mapped to its part-of-speech tag in the context of a given sentence, context-sensitive spelling correction where a word is mapped into a similarly spelled word that is appropriate in the context of the sentence, and many other problems like word-sense disambiguation, word choice selection in machine translation and identifying discourse markers.

Similarly, we define Question Classification (QC) here to be the multi-class classification task that seeks a mapping $g : X \rightarrow \{c_1, ..., c_n\}$ from an instance $x \in X$ (e.g., a question) to one of n classes $c_1, ..., c_n$. This classification provides a semantic constraint on the sought-after answer. The intention is that this classification, potentially with other constraints on the answer, will be used by a downstream process to select a correct answer from several candidates.

Earlier works have suggested various standards of classifying questions. Wendy Lehnert's conceptual taxonomy (Lehnert 1986), for example, proposes about 13 conceptual classes including *causal antecedent, goal orientation, enablement, causal consequent, verification, disjunctive*, and so on. However, in the context of factual questions that are of interest to us here, conceptual categories do not seem to be helpful; instead, our goal is to *semantically* classify questions, as in some earlier work (Harabagiu et al. 2000; Singhal et al. 2000; Hermjakob 2001). The key difference, though, is that we attempt to do that with a significantly finer taxonomy of answer types; the hope is that with the semantic answer types as input, one can easily locate answer candidates, given a reasonably accurate named entity recognizer for documents.

For example, in the next two questions, knowing that the targets are a **city** or a **country** will be more useful than just knowing that they are **locations**.

Q: *What Canadian city has the largest population?*
Q: *Which country gave New York the Statue of Liberty?*

2.1 Question Hierarchy

We define a two-layered taxonomy, which represents a natural semantic classification for typical answers. The hierarchy contains 6 coarse classes (ABBREVIATION, DESCRIPTION, ENTITY, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine classes. Table 1 demonstrates the distribution of these classes in the $1,000$

questions taken from TREC (Text Retrieval Conference (Voorhees 2002)) 10 and 11, used for our experimental evaluation. Each coarse class contains a non-overlapping set of fine classes. The motivation behind adding a level of coarse classes is that of compatibility with previous work's definitions, and comprehensibility.

Class	#	Class	#
ABBREVIATION	18	term	19
abbreviation	2	vehicle	7
expression	16	word	0
DESCRIPTION	153	**HUMAN**	171
definition	126	group	24
description	13	individual	140
manner	7	title	4
reason	7	description	3
ENTITY	174	**LOCATION**	195
animal	27	city	44
body	5	country	21
color	12	mountain	5
creative	14	other	114
currency	8	state	11
disease/medicine	3	**NUMERIC**	289
event	6	code	1
food	7	count	22
instrument	1	date	146
lang	3	distance	38
letter	0	money	9
other	19	order	0
plant	7	other	24
product	9	period	18
religion	1	percent	7
sport	3	speed	9
substance	20	temp	7
symbol	2	vol.size	4
technique	1	weight	4

Table 1. *Distribution of* $1,000$ *TREC questions over the question hierarchy. Coarse classes are in bold and are followed by their refinements into fine classes. # is the number of questions in each class. The questions were manually classified by us.*

2.2 The Ambiguity Problem

One difficulty in the question classification task is that there is no completely clear boundary between classes. Therefore, the classification of a specific question according to our class hierarchy can still be ambiguous although we have tried to define it as clearly as possible. Consider questions:

1. *What is bipolar disorder?*
2. *What do bats eat?*
3. *What is the PH scale?*

Question 1 could belong to **definition** or **disease/medicine**; Question 2 could belong to **food**, **plant** or **animal**; and Question 3 could be a **NUMERIC:other** (nontypical numeric value) or a **definition**. It is hard to categorize **these** questions into one single class and it is likely that mistakes will be introduced in the downstream process if we do so. To avoid this problem, we allow our classifiers to assign multiple class labels for a single question in the question answering system. This strategy is better than only allowing one label because we can apply all the classes in the later precessing steps without loss. For the purpose of evaluation, however, only the top-ranked coarse and fine class are counted as correct.

3 Learning a Question Classifier

In our work, a question can be mapped to one of 6 possible coarse classes and one of 50 fine classes (We call the set of possible class labels for a given question a *confusion set*).

One difficulty in supporting fine-grained classification of this level is the need to extract from the questions finer features that require syntactic and semantic analysis of questions. As a result, existing non-learning approaches, as in (Singhal et al. 2000), have adopted a small set of simple answer entity types, which consist of the classes: *Person, Location, Organization, Date, Quantity, Duration, Linear_Measure*. Some of the rules used by them in classification were of the following forms:

- If a query starts with *Who* or *Whom*: type **Person**.
- If a query starts with *Where*: type **Location**.
- If a query contains *Which* or *What*, the head noun phrase determines the class, as for What X questions.

Although the manual rules may have large coverage and reasonable accuracy over their own taxonomy, confined by the tedious work on analyzing a large number of questions and the requirements of an explicit construction and representation of the mapping from questions to classes, most earlier question answering systems, therefore, can only perform a coarse classification for no more than a fairly small set (e.g. 20 classes). It is not sufficient to support fine-grained classification, nor to handle an even larger set of question types that we can anticipate in an interactive scenario.

On the contrary, learning technologies can solve these difficulties easily. In our learning approach, one can define only a small number of 'types' of features based on previous syntactic and semantic analysis results, which are then expanded in a data-driven way to a potentially large number of features, relying on the ability of the learning process to handle it. In addition to this advantage, a learned classifier is more flexible to reconstruct than a manual one because it can be trained on a new taxonomy in a very short time.

3.1 A Hierarchical Classifier

To adapt to the layered semantic hierarchy of answer types, we develop a hierarchical learning classifier based on the sequential model of multi-class classification, as described in (Even-Zohar and Roth 2001). The basic idea in this model is to reduce the set of candidate class labels for a given question step by step by concatenating a sequence of simple classifiers. The output of one classifier - a set of class labels - is used as input to the next classifier. In order to allow a simple classifier to output more than one class label in each step, the classifier's output activation is normalized into a density over the class labels and is thresholded.

The question classifier is built by combining a sequence of two simple classifiers. The first classifies questions into coarse classes (*Coarse Classifier*) and the second into fine classes (*Fine Classifier*). Each of them utilizes the Winnow algorithm within the SNoW (Sparse Network of Winnows (Carlson et al. 1999)) learning architecture. SNoW is a multi-class learning architecture that is specifically tailored for large scale learning tasks (Roth 1998). It learns a separate linear function over the features for each class label based on a feature efficient learning algorithm, Winnow (Littlestone 1989). It is suitable for learning in NLP-like domains and robust to a large feature space where the number of potential features is very large, but only a few of them are active in each example, and only a small fraction of them are relevant to the target concept.

A feature extractor automatically extracts the same features for both classifiers based on multiple syntactic, semantic analysis results and external knowledge of the question. The second classifier depends on the first in that its candidate labels are generated by expanding the set of retained coarse classes from the first into a set of fine classes; this set is then treated as the confusion set for the second classifier. Figure 1 shows the basic structure of the hierarchical classifier. During either the training or the testing stage, a question is processed along one single path top-down to get classified.

The detailed classification process can be formally explained by the following scenario: The initial confusion set of any question q is $C_0 = \{c_1, c_2, \ldots, c_n\}$, the set of all the coarse classes. The coarse classifier determines a set of preferred labels, $C_1 = Coarse\_Classifier(C_0, q)$, $C_1 \subseteq C_0$ so that $|C_1| \leq 5$ (5 is chosen through experiments). Then each coarse class label in C_1 is expanded to a fixed set of fine classes determined by the class hierarchy. That is, suppose the coarse class c_i is mapped into the set $c'_i = \{f_{i1}, f_{i2}, \ldots, f_{im}\}$ of fine classes, then $C_2 = \bigcup_{c_i \in C_1} c'_i$. The fine classifier determines a set of preferred labels, $C_3 = Fine\_Classifier(C_2, q)$ so that $C_3 \subseteq C_2$ and $|C_3| \leq 5$. C_1 and C_3 are the ultimate outputs from the whole classifier.

3.2 Decision Model

For both the coarse and fine classifiers, the same decision model is used to choose class labels for a question. Given a confusion set and a question, SNoW outputs a density over the classes derived from the activation of each class. After ranking the

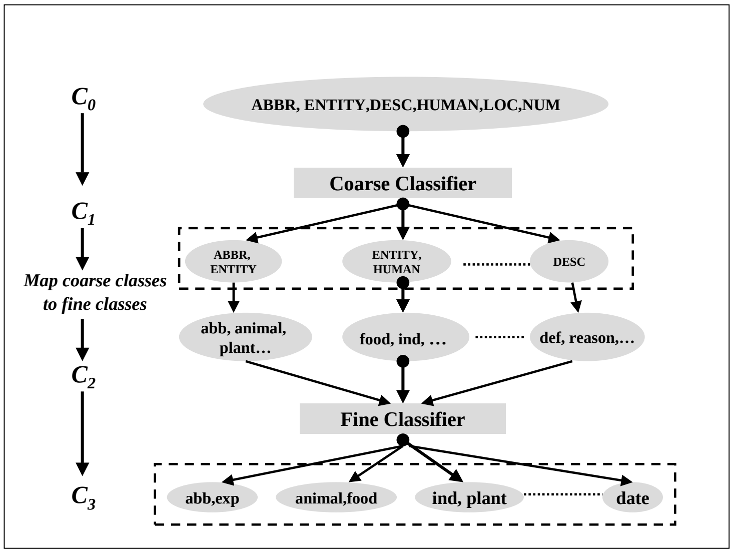


Fig. 1. The hierarchical classifier

classes in the decreasing order of density values, we have the possible class labels $C = \{c_1, c_2, \ldots, c_n\}$, with their densities $P = \{p_1, p_2, \ldots, p_n\}$ (where, $\sum_{i=1}^{n} p_i = 1$, $0 \le p_i \le 1$, $1 \le i \le n$). As discussed earlier, for each question we output the first k classes ($1 \le k \le 5$), $c_1, c_2, \ldots c_k$ where k satisfies,

$$k = min(argmin_t(\sum_{i=1}^{t} p_i \ge T), 5) \tag{1}$$

T is a threshold value in $[0, 1]$ and $T = 0.95$ is chosen through experiments. If we treat p_i as the probability that a question belongs to class i, the decision model yields a reasonable probabilistic interpretation. We are 95 percent sure that the correct label is inside those k classes.

4 Features in Question Classification

Machine Learning based classifiers typically take as input a feature-based representation of the domain element (e.g., a question). For the current task, a question sentence is represented as a vector of features and treated as a training or test example for learning. The mapping from a question to a class label is a linear function defined over this feature vector.

In addition to the information that is readily available in the input instance, it is common in natural language processing tasks to augment sentence representation with syntactic categories – part-of-speech (POS) and phrases, under the assumption that the sought-after property, for which we seek the classifier, depends on the syntactic role of a word in the sentence rather than the specific word (Roth

1998). Similar logic can be applied to semantic categories. In many cases, the property does not seem to depend on the specific word used in the sentence – that could be replaced without affecting this property – but rather on its 'meaning' (Li, Small, and Roth 2004). For example, given the question: `What Cuban dictator did Fidel Castro force out of power in 1958?`, we would like to determine that its answer should be a name of a person. Knowing that `dictator` refers to a person is essential to correct classification.

In this work, several primitive feature types are derived from multiple sources of syntactic and lexical semantic analysis of questions, each of which in itself could be a learning process, described later in this section. Over these primitive feature types, a set of operators is used to compose more complex features, such as conjunctive (n-grams) and relational features. A simple script that describes the 'types' of features used, (e.g., conjunction of two consecutive words and their POS tags) is written and the features themselves are extracted in a data driven way. Only 'active' features (that is, the binary features with 'true' values in the current example) are listed in our representation so that despite the large number of potential features, the size of each example is small.

The learning architecture we use allows a multi-class classifier to handle a relatively huge feature space (in our case, the dimension of the feature space is above $200,000$), laying a foundation for the feature-enriched classification strategy.

4.1 Syntactic Features

In addition to *words*, the syntactic features for each question include *POS tags*, *chunks* (non-overlapping phrases in a sentence as defined in (Abney1991)), *head chunks* (e.g., the first noun chunk and the first verb chunk after the question word in a sentence).

Part-of-speech information of the words in a question is annotated by a POS tagger (Even-Zohar and Roth 2001) that also makes use of the sequential model to restrict the number of competing classes (POS tags) while maintaining, with high probability, the presence of the true outcome in the candidate set. It achieves state–of–the–art results on this task and is more efficient than other part-of-speech taggers. *Chunks* and *head chunks* of a question are extracted by a publicly available shallow parser described in (Punyakanok and Roth 2001) [2]. The preference of shallow processing over full parsing is due to the consideration of the potential application of question answering in an interactive environment which requires high robustness with noisy input. The following example illustrates the information available when generating the syntax-augmented feature-based representation.

Question: *Who was the first woman killed in the Vietnam War?*

POS tagging: *[Who WP] [was VBD] [the DT] [first JJ] [woman NN] [killed*

[2] All of these tools are freely available at http://L2R.cs.uiuc.edu/∼cogcomp/.

VBN] [in IN] [the DT] [Vietnam NNP] [War NNP] [? .]

Chunking: *[NP Who] [VP was] [NP the first woman] [VP killed] [PP in] [NP the Vietnam War] ?*

The head chunks denote the first noun and the first verb chunk after the question word in a question. For example, in the above question, the first noun chunk after the question word *Who* is 'the first woman'.

4.2 Semantic Features

Similarly to syntactic information like part-of-speech tags, a fairly clear notion of how to use lexical semantic information is: we replace or augment each word by its semantic class in the given context, generate a feature-based representation, and then learn a mapping from this representation to the sought-after property. This general scheme leaves open several issues that make the analogy to syntactic categories non-trivial. First, it is not clear what the appropriate semantic categories are and how to acquire them. Second, it is not clear how to handle the more difficult problem of semantic disambiguation when augmenting the representation of a sentence.

For the first problem, we study several lexical semantic information sources that vary in their granularity, the difficulty to acquire them and the accuracy within which they are acquired. The information sources are: (1) named entities, (2) word senses in WordNet (Fellbaum 1998), (3) manually constructed word lists related to specific answer types and (4) automatically-generated semantically similar words for every common English word based on distributional similarity. All the sources are acquired by external semantic analysis tools.

For the second problem above, in all cases, we define semantic categories of words and incorporate the information into question classification in the same way: if a word w occurs in a question, the question representation is augmented with the semantic category(ies) of the word. For example, in the question: *What is the state flower of California ?* given that *plant* (say) is the only semantic class of flower, the feature extractor adds *plant* to the question representation.

Clearly, a word may belong to different semantic categories in different contexts. For example, the word *water* has the meaning *liquid* or *body of water* in different sentences. Without disambiguating the sense of a word we cannot determine which semantic category is more appropriate in a given context. At this point, our solution is to extract all possible semantic categories of a word as features, without disambiguation, and allowing the learning process to deal with this problem, building on the fact that the some combinations of categories are more common than others and more indicative to a specific class label. As we show later, our experiments support this decision, although we have yet to experiment with the possible contribution of a better way to determine the semantic class in a context sensitive manner.

Named Entities

A named entity (NE) recognizer assigns a semantic category to some of the noun phrases in the question. The scope of the categories used here is broader than the common named entity recognizer. With additional categories such as *profession*, *event*, *holiday*, *plant*, *sport*, *medical* etc., we redefine our task in the direction of semantic categorization. The named entity recognizer was built on the shallow parser described in (Punyakanok and Roth 2001), and was trained to categorize noun phrases into one of 34 different semantic categories of varying specificity. Its overall accuracy ($F_{\beta=1}$) is above 90 percent. For the question *Who was the woman killed in the Vietnam War ?*, the named entity tagger will get: **NE:** *Who was the [Num first] woman killed in the [Event Vietnam War] ?* As described above, the identified named entities are added to the question representation.

WordNet Senses

In WordNet (Fellbaum 1998), words are organized according to their 'senses' (meanings). Words of the same sense can, in principle, be exchanged in some contexts. The senses are organized in a hierarchy of hypernyms and hyponyms. Word senses provide another effective way to describe the semantic category of a word. For example, in WordNet 1.7, the word *water* belongs to five senses. The first two senses are:

Sense 1: binary compound that occurs at room temperature as a colorless odorless liquid;

Sense 2: body of water.

Sense 1 contains words {H2O, water} while Sense 2 contains {water, body of water}. Sense 1 has a hypernym (**Sense 3**: binary compound); and Sense 2 has a hyponym (**Sense 4: tap water**).

For each word in a question, all of its sense IDs and direct hypernym and hyponym IDs are extracted as features.

Class-Specific Related Words

Each question class frequently occurs together with a set of words which can be viewed as semantically related to this class. We analyzed about 5,500 questions and manually extracted a list of related words for each question class. These lists are different from ordinary named entity lists in a way that they cross the boundary of the same syntactic role. Below are some examples of the word lists.

Question Class: Food

{*alcoholic apple beer berry breakfast brew butter candy cereal champagne cook delicious eat fat feed fish flavor food fruit intake juice pickle pizza potato sweet taste ...*}

Question Class: Mountain

{*hill ledge mesa mountain peak point range ridge slope tallest volcanic volcano...*}

The question class can be viewed as a ‘topic’ tag for words in the list, a type of semantic categories. It is a semantic information source similar to the keyword information used in some earlier work (Harabagiu et al. 2000; Hermjakob 2001). The difference is that they are converted into features here and combined with other types of features to generate an automatically learned classifier.

Distributional Similarity Based Categories

The distributional similarity (Lee 1999) of words captures the likelihood of them occurring in identical syntactic structures in sentences. Depending on the type of dependencies used to determine the distributional similarity, it can be argued that words with high distribution similarity have similar meanings. For example, the words used in the following syntactic structures are likely to be U.S. states.

... ’s appellate court
... ’s capital
... ’s driver’s license
... ’s sales tax
campaign in ...
governor of ...
illegal in ...
senator for ...

Pantel and Lin (Pantel and Lin 2002) proposed a method to cluster words into semantically similar groups based on their distributional similarity with respect to a large number of dependencies. They built similar word lists for over 20, 000 English words. All the words in a list corresponding to a target word are organized into different senses. For example, the word *water* has the following similar words:

Sense 1: {*oil gas fuel food milk liquid ...*}
Sense 2: {*air moisture soil heat area rain snow ice ...*}
Sense 3: {*waste sewage pollution runoff pollutant...*}

One way to apply these lists in question classification is to treat the target word (in the above example, ‘water’) of a list as the semantic category of all the words in the list and in line with our general method, and add this semantic category of the word as a feature.

5 Experimental Study

Our experimental study focuses on (1) testing the performance of the learned classifier in classifying factual questions into coarse and fine classes, and (2) comparing the contribution of different syntactic and semantic features to the classification quality.

Based on the same framework of the hierarchical classifier described before, we construct different classifiers utilizing different feature sets and compare them in experiments. The first group of classifiers compared, take as input an incremental combination of syntactic features (words, POS tags, chunks and head chunks). In particular, the classifier that takes as input all the syntactic features is denoted as SYN. Then, another group of classifiers are constructed by adding different combinations of semantic features such as NE – named entity features, SemWN – features from WordNet senses, SemCSR – features based on class-specific words and SemSWL – semantically similar word lists, to the input of the SYN classifier.

Three experiments are conducted for the above purposes. The first evaluates the accuracies of the hierarchical classifier for both coarse and fine classes using only syntactic features. The second evaluates the contribution of different semantic features (all 15 possible combinations of semantic feature types are added to the SYN classifier and compared this way.). In the third experiment we hope to find out the relation between the contribution of semantic features and the size of the training set by training the classifier with training sets of different sizes.

The 1000 questions taken from TREC (Voorhees 2002) 10 and 11 serve as an ideal test set for classifying factual questions. $21,500$ training questions are collected from three sources: 894 TREC 8 and 9 questions, about 500 manually constructed questions for a few rare classes, and questions from the collection published by USC (Hovy et al. 2001). In the first two experiments, the classifiers are trained on all these questions. 10 other training sets with incremental sizes of $2,000$, $4,000$, ..., $20,000$ questions built by randomly choosing from these questions are used in the third experiment. All the above questions were manually labelled according to our question hierarchy, with one label per question according to the majority of our annotators. All the of above data sets are available at http://l2r.cs.uiuc.edu/∼cogcomp/.

Performance is evaluated by the global accuracy of the classifiers for all the coarse or fine classes (Accuracy), and the accuracy of the classifiers for a specific class c (Precision[c]), defined as follows:

$$Accuracy = \frac{\#\ of\ correct\ predictions}{\#\ of\ predictions} \tag{2}$$

$$precison[c] = \frac{\#\ of\ correct\ predictions\ of\ class\ c}{\#\ of\ predictions\ of\ class\ c} \tag{3}$$

Note that since all questions are being classified, the global accuracy is identical to both precision and recall that are commonly used in similar experiments. However, for specific classes, precision and recall are different because questions of one class can be predicted as belonging to another. We only show $precision[c]$ for each class c in Table 3) since high accuracy on all classes implies high recall for each specific class.

Although we allow the decision model to output multiple class labels in each step for practical application, only one coarse class and one fine class which are ranked the first by their density values in C_1 and C_3 are counted as correct in evaluation.

5.1 Experimental Results

All the classifiers are trained on the 21, 500 training questions and tested on the 1, 000 TREC (Voorhees 2002)) 10 and 11 questions in the experiments except the case of studying the influence of training sizes.

Classification Performance Using Only Syntactic Features

Table 2 shows the classification accuracy of the hierarchical classifier with different sets of syntactic features in the first experiment. Word, POS, Chunk and Head(SYN) represent different feature sets constructed from an incremental combination of syntactic features (for example, the feature set Chunk actually contains all the features in Word, POS and also adds chunks, and Head(SYN) contains all the four types of syntactic features.). Overall, we get a 92.5 percent accuracy for coarse classes and 85 percent for the fine classes using all the syntactic features. The reason for the lower performance in classifying fine classes compared with the performance on coarse classes is because there are far more fine classes and because they have less clear boundaries. Although chunks do not seem to contribute to the classification quality in the experiment using the feature set Chunk, they contribute to it when combined with head chunks as in Head(SYN). The fact that head chunk information contributes more than generic chunks indicates that the syntactic role of a chunk is a factor that can not be ignored in this task.

Classifier	Word	POS	Chunk	Head(SYN)
Coarse	85.10	91.80	91.80	**92.50**
Fine	82.60	84.90	84.00	**85.00**

Table 2. *Classification Accuracy of the hierarchical classifier for coarse and fine classes using an incremental combination of syntactic features.*

Contribution of Semantic Features

Although only minor improvements are acquired (not shown) in classifying questions into coarse classes after semantic features are added, significant improvements are achieved for distinguishing between fine classes. Figure 2 presents the accuracy of the classifier for fine classes after semantic features are input together with the SYN feature set.

The best accuracy (89.3 percent) for classifying fine classes in this experiment is achieved using a combination of feature types {SYN, NE, SemCSR, SemSWL}. This is a 28.7 percent error reduction (from 15 percent to 10.7 percent) over the SYN classifier. For simplicity, this feature set {SYN, NE, SemCSR, SemSWL} is

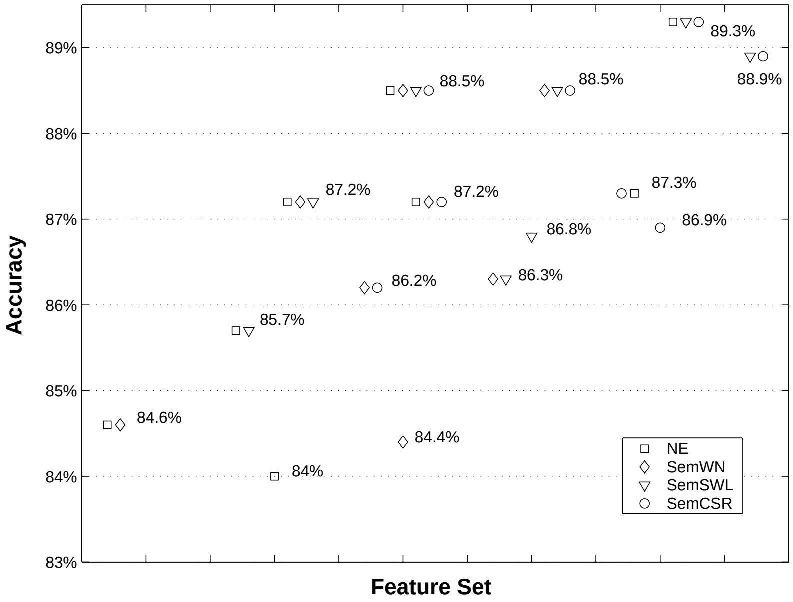


Fig. 2. Classification Accuracy for fine classes after adding different combinations of semantic features to the input of the SYN classifier. Shapes in the graph represent the four types of semantic feature {NE, SemWN, SemCSR, SemSWL} and a juxtaposition of symbols represents the use of a combination of different types(in addition to SYN). For example, ▽○ denotes that the classifier takes as input a combination of feature types {SYN, SemCSR, SemSWL}.

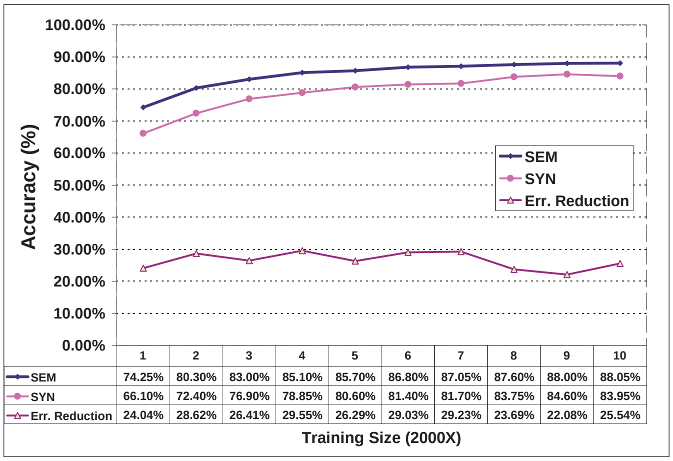
denoted as 'SEM' in the later experiments. The results reflect that lexical semantic information has contributed much to fine-grained classification, even without word sense disambiguation. Furthermore, it takes only about 30 minutes to train the SEM classifier over 20,000 questions, an indication of the efficiency of the SNoW learning algorithm.

However, the performance of using all features types is only 88.5 percent. Although WordNet features may contribute to the classification quality by itself, it hurts when combined with all semantic feature types. This is probably due to the fact that WordNet features may contribute overlapping information of other feature types but add more noise. It also indicates that while the number and type of features are important to the classification quality, using a learning algorithm that can tolerate a large number of features is also important. In this experiment we also noticed that the class-specific word lists (SemCSR), and similar word lists (SemSWL) are the most beneficial sources of semantic information.

Classification Performance vs. Training Size

The relation between classification accuracy of the SYN classifier and the SEM classifier, and training size, is tested in the third experiment and results are given in Figure 3. The error reduction from the SYN classifier to the SEM classifier on the

1, 000 TREC questions is stable over 20 percent over all training sizes, also proving the distinctive contribution of the semantic features in this task.



	1	2	3	4	5	6	7	8	9	10
SEM	74.25%	80.30%	83.00%	85.10%	85.70%	86.80%	87.05%	87.60%	88.00%	88.05%
SYN	66.10%	72.40%	76.90%	78.85%	80.60%	81.40%	81.70%	83.75%	84.60%	83.95%
Err. Reduction	24.04%	28.62%	26.41%	29.55%	26.29%	29.03%	29.23%	23.69%	22.08%	25.54%

Fig. 3. Classification Accuracy versus training size. 'SYN' and 'SEM' represent the learning curves of the SYN classifier and the SEM classifier respectively. 'Err. Reduction' denotes the error reduction from the SYN classifier to the SEM classifier. The training size is $2000 \times X$ and the test set is $1,000$ TREC questions.

5.2 Further Analysis

Some other interesting phenomena have also been observed in our experiments. The classification accuracy of the SEM classifier for specific fine classes is given in Table 3. It is shown in the graph that the accuracies for specific fine classes are far from uniform, reflecting difference of classification difficulty. Questions belonging to **desc** (description) and **Entity:other** (uncommon entities) are the most difficult to identify among all fine classes, since their boundaries with other classes are quite fuzzy.

A specific metric is defined to evaluate the overlapping degree of question classes. The tendency that class i is confused with class j (D_{ij}) is defined as follows:

$$D_{ij} = 2 \cdot Err_{ij}/(N_i + N_j) \tag{4}$$

If we return exactly one label for each question, Err_{ij} is the number of questions in class i misclassified as class j. N_i and N_j are the numbers of questions in class i and j separately. Figure 4 is a gray-scale map of the matrix $D[n, n]$. $D[n, n]$ is so sparse that most parts of the graph are blank. From this graph, we can see that there is no good clustering property among the fine classes inside a coarse class.

Class	**#**	**Precision[c]**	**Class**	**#**	**Precision[c]**
abb	2	100%	desc	25	36%
exp	17	94.11%	manner	8	87.5%
animal	27	85.18%	reason	7	85.71%
body	4	100%	gr	19	89.47%
color	12	100%	ind	154	90.25%
cremat	13	76.92%	title	4	100%
currency	6	100%	desc	3	100%
dismed	4	50%	city	41	97.56%
event	4	75%	country	21	95.23%
food	6	100%	mount	2	100%
instru	1	100%	LOC:other	116	89.65%
lang	3	100%	state	14	78.57%
ENTY:other	24	37.5%	count	24	91.66%
plant	3	100%	date	145	100%
product	6	66.66%	dist	37	97.29%
religion	1	100%	money	6	100%
sport	4	75%	NUM:other	15	93.33%
substance	21	80.95%	period	20	85%
symbol	2	100%	perc	9	77.77%
termeq	22	63.63%	speed	8	100%
veh	7	71.42%	temp	4	100%
def	125	97.6%	weight	4	100%
TOTAL	**1000**	**89.3%**			

Table 3. *Classification Accuracy for specific fine classes with the feature set SEM. # denotes the number of predictions made for each class and Precision[c] denotes the classification accuracy for a specific class c. The classes not shown do not actually occur in the test collection.*

To better understand the classification results, we also split the 1,000 test questions into different groups according to their question words, that is, *What*, *Which*, *Who*, *When*, *Where*, *How* and *Why* questions. A baseline classifier, Wh-Classifier, is constructed by classifying each group of questions into its most typical fine class. Table 4 shows the *accuracy* (defined as $\frac{\#\ of\ correct\ predicted\ questions}{\#\ of\ test\ questions}$) of the Wh-Classifier and the SEM classifier on different groups of questions. The typical fine classes in each group and the number of questions in each class are also given. The distribution of *What* questions over the semantic classes is quite diverse, and therefore, they are more difficult to classify than other groups.

From this table, we also observe that classifying questions just based on question words (1) does not correspond well to the desired taxonomy, and (2) is too crude since a large fraction of the questions are ‘What questions’.

The overall accuracy of our learned classifier is satisfactory. Indeed, all the reformulation questions that we exemplified at the beginning of this paper have been

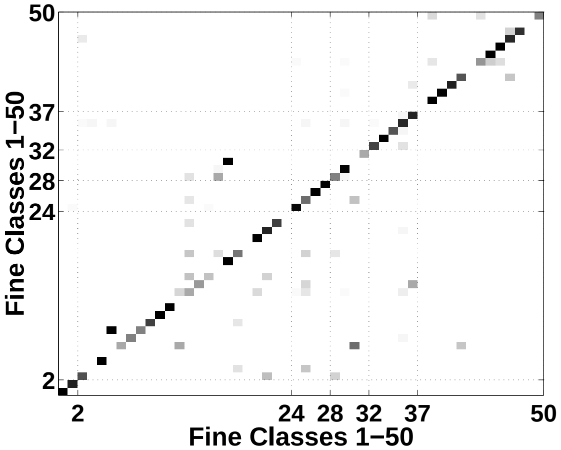


Fig. 4. The gray–scale map of the matrix $D[n, n]$. The gray scale of the small box in position (i, j) denotes D_{ij}. The larger D_{ij} is, the darker the box is. The dotted lines separate the 6 coarse classes.

Question Word	**#**	**Wh**	**SEM**	**Classes(#)**
What	598	21.07%	85.79%	ind.(36), def.(126), loc–other(47)
Which	21	33.33%	95.24%	ind.(7), country(5)
Who	99	93.94%	96.97%	group(3), ind.(93), human desc.(3)
When	96	100%	100%	date(96)
Where	66	90.01%	92.42%	city(1), mount.(2), loc–other(61)
How	86	30.23%	96.51%	count(21), dist.(26), period(11)
Why	4	100%	100%	reason(4)
Total	1000	41.3%	89.3%	

Table 4. *Classification Accuracy of the Wh-Classifier and the SEM classifier on different question groups. Typical fine classes in each group and the number of questions in each class are also shown by Classes(#).*

correctly classified. Nevertheless, it is constructive to consider some cases in which the classifier fails. Below are some examples misclassified by the SEM classifier.

- *What imaginary line is halfway between the North and South Poles ?*

The correct label is **location**, but the classifier outputs an arbitrary class. Our classifier fails to determine that ‘line’ might be a location even with the semantic information, probably because some of the semantic analysis is not context sensitive.

- *What is the speed hummingbirds fly ?*

The correct label is **speed**, but the classifier outputs **animal**. Our feature extractor fails to determine that the focus of the question is ‘speed’. This example illustrates the necessity of identifying the question focus by analyzing syntactic structures.

- *What do you call a professional map drawer ?*

The classifier returns **other entities** instead of **equivalent term**. In this case, both classes are acceptable. The ambiguity causes the classifier not to output **equivalent term** as the first choice.

6 Related Work

In an earlier work (Pinto et al. 2002), a simple question classification system is constructed based on language models. More recent works address the question classification problem using more involved machine learning techniques include (Radev et al. 2002), (Hacioglu and Ward 2003) and (Zhang and Lee 2003). (Radev et al. 2002) defines a smaller taxonomy and applies the Rappier rule leaning algorithm with a lot fewer features. (Zhang and Lee 2003) compares several learning algorithms for question classification using the taxonomy developed in an early version of the work presented here (Li and Roth 2002) and have shown that Support Vector Machine (SVM) with a tree kernel can achieve performance improvement over a single-layer SNoW classifier using the same primitive syntactic features. This is expected, since using tree kernels is equivalent to enriching the feature space with conjunction features. However, the goal of the work presented here is to show that a sensible incorporation of semantic features can improve the quality of question classification significantly.

7 Conclusion and Future Directions

This paper presents a machine learning approach to question classification, modeled as a multi-class classification task with 50 classes. We developed a hierarchical classifier that is guided by a layered semantic hierarchy of answers types, and used it to classify questions into fine-grained classes. Our experimental results show that the question classification problem can be solved quite accurately (nearly 90 percent accuracy) using a learning approach, and exhibit the benefits of an enhanced feature representation based on lexical semantic analysis. While the contribution of syntactic information sources to the process of learning classifiers has been well studied, we hope that this work can inspire the systematic studies of the contribution of semantic information to classification.

In an attempt to compare the four semantic information sources, Table 5 presents the average number of semantic features extracted for a test question in each case. This gives some indication for the amount of information (in some sense, that is also the noise level) added by each of the sources. Among the four semantic information sources, named entity recognition is the only context sensitive semantic analysis of words. All the other three sources add noise to the representation of a question due to lack of sense disambiguation.

However, confined by the insufficient coverage of semantic categories and words, and also the recognition accuracy, named entities contribute the least to the classification. On the contrary, the class-specific word lists (SemCSR), and similar word lists (SemSWL) have much larger coverage and SemCSR has a more direct connection between words and question classes. Although we cannot get to the conclusion that the noise does not degrade the performance in the learning process, clearly the coverage is a more important factor in deciding the classification quality — another evidence of the advantage of learning in classification.

Feature Type	avg. # of features
NE	0.23
SemWN	16
SemCSR	23
SemSWL	557

Table 5. *The average number of semantic features extracted for each test question based on different types of semantic features. For example, there are* 16 *SemWN features extracted for each question on average.*

The question classifier introduced in this paper has already been incorporated into our practical question answering system (Roth et al. 2002) to provide wide support to later processing stages, such as passage retrieval and answer selection and verification. We hope to evaluate quantitively the contribution of the question classifiers to this system when it reaches a relatively mature status. Another step in this line of work would be to improve the selection of the semantic classes using context sensitive methods for most of the semantic information sources and to enlarge the coverage of the named entity recognizer. The third direction is to incorporate question classes with other analysis results to form an abstract representation of question information, providing comprehensive constraints over possible answers. Furthermore, we hope to extend this work to support interactive question answering. In this task, the question answering system could be able to interact with users to lead to more variations of questions but with more contextual information. It may require even larger coverage of semantic classes and more robustness, and a strategy that is more adaptive to the answer selection process.

References

Abney, S. P. 1991. Parsing by chunks. In S. P. Abney R. C. Berwick and C. Tenny, editors, *Principle-based parsing: Computation and Psycholinguistics*. Kluwer, Dordrecht, pages 257–278.

Carlson, A., C. Cumby, J. Rosen, and D. Roth. 1999. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May.

Even-Zohar, Y. and D. Roth. 2001. A sequential model for multi-class classification. In *Proceedings of EMNLP-2001, the SIGDAT Conference on Empirical Methods in Natural Language Processing*, pages 10–19.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database.* The MIT Press, May.

Hacioglu, K. and W. Ward. 2003. Question classification with support vector machines and error correcting codes. In *Proceedings of HLT-NAACL.*

Harabagiu, S., D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. 2000. Falcon: Boosting knowledge for answer engines. In E. Voorhees, editor, *Proceedings of the 9th Text Retrieval Conference, NIST*, pages 479–488.

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

Hirschman, L., M. Light, E. Breck, and J. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 325–332.

Hovy, E., L. Gerber, U. Hermjakob, C. Lin, and D. Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the DARPA HLT conference.*

Lee, L. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32.

Lehnert, W. G. 1986. A conceptual theory of question answering. In B. J. Grosz, K. Sparck Jones, and B. L. Webber, editors, *Natural Language Processing.* Kaufmann, Los Altos, CA, pages 651–657.

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

Li, X., K. Small, and D. Roth. 2004. The role of semantic information in learning question classifiers. In *Proceedings of the First Joint International Conference on Natural Language Processing.*

Littlestone, N. 1989. *Mistake bounds and logarithmic linear-threshold learning algorithms.* Ph.D. thesis, U. C. Santa Cruz, March.

Moldovan, D., M. Pasca, S. Harabagiu, and M. Surdeanu. 2002. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 33–40.

Pantel, P. and D. Lin. 2002. Discovering word senses from text. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Pinto, D., M. Branstein, R. Coleman, M. King, W. Li, X. Wei, and W.B. Croft. 2002. Quasm: A system for question answering using semi-structured data. In *Proceedings of the Joint Conference on Digital Libraries.*

Punyakanok, V. and D. Roth. 2001. The use of classifiers in sequential inference. In *Proceedings of the 13th Conference on Advances in Neural Information Processing Systems*, pages 995–1001. MIT Press.

Radev, D. R., W. Fan, H. Qi, H. Wu, and A. Grewal. 2002. Probabilistic question answering from the web. In *Proceedings of WWW-02, 11th International Conference on the World Wide Web.*

Roth, D. 1998. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of 15th National Conference on Artificial Intelligence (AAAI).*

Roth, D., C. Cumby, X. Li, P. Morie, R. Nagarajan, N. Rizzolo, K. Small, and W. Yih. 2002. Question answering via enhanced understanding of questions. In *Proceedings of the 11th Text Retrival Conference, NIST*, pages 592–601.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Singhal, A., S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira. 2000. AT&T at TREC-8. In E. Voorhees, editor, *Proceedings of the 8th Text Retrieval Conference, NIST.*

Voorhees, E. 2002. Overview of the TREC-2002 question answering track. In *Proceedings of the 11th Text Retrieval Conference, NIST*, pages 115–123.

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.