
Information Retrieval, 8, 481–504, 2005


# Analysis of Statistical Question Classification for Fact-Based Questions

DONALD METZLER
W. BRUCE CROFT
*University of Massachusetts, Amherst*

**Abstract.** Question classification systems play an important role in question answering systems and can be used in a wide range of other domains. The goal of question classification is to accurately assign labels to questions based on expected answer type. Most approaches in the past have relied on matching questions against hand-crafted rules. However, rules require laborious effort to create and often suffer from being too specific. Statistical question classification methods overcome these issues by employing machine learning techniques. We empirically show that a statistical approach is robust and achieves good performance on three diverse data sets with little or no hand tuning. Furthermore, we examine the role different syntactic and semantic features have on performance. We find that semantic features tend to increase performance more than purely syntactic features. Finally, we analyze common causes of misclassification error and provide insight into ways they may be overcome.

**Keywords:** question classification, question answering, machine learning, Support Vector Machines, syntactic features, semantic features, WordNet

## 1. Introduction

### *1.1. Overview*

Question classification is the process by which a system analyzes a question and labels the question based on its expected answer type. For example, the question "*Who was the first Prime Minister of Canada?*" expects a person's name as an answer. Given a finite set of possible expected answer types, known as a *question ontology*, the goal of a question classification system is to learn a mapping from questions to answer types. Although this task may sound simple, there are many factors that determine how well such systems perform and how robust they are. This paper highlights and analyzes these factors in a statistical machine learning framework.

We focus our attention on *fact-based* questions. These questions are typically pointed, trivia-like questions where a short, factual answer is expected. Examples of such questions are: "*Where is the Orinoco River?*", "*What type of currency is used in Australia?*", and "*What is the speed of light?*". Although interesting, other types of questions, such as *task-oriented* questions, are not explored.

Question classification systems are primarily used as components of *question answering* (QA) systems. QA is the task of retrieving answers to questions posed in natural language from a collection of documents, where an answer is generally a short fragment of text drawn from the corpus. QA systems are a shift away from classical *document*

*retrieval* towards *information retrieval*. This saves the user valuable time by eliminating the need to search through a long ranked list of documents for an answer to their question.

There are many kinds of QA systems, all with different underlying architectures. Although each system varies in the way it produces an answer to a given question, most systems follow a general framework (Voorhees 2001). Given a question, most systems first analyze the question and use a question classification system to determine the most likely expected answer type or types. Next, some form of document or passage-level retrieval is done to retrieve candidate answers from the corpus. Finally, the named entities within the retrieved documents/passages are tagged. This allows the system to prune possible answers based on the expected answer type(s) returned by the question classification system. For instance, if the expected answer type is most likely a person, then only those documents/passages that contain person entity tags are considered possible answers. From this list of candidates, the system determines the best answer or list of answers to present to the user. If the original classification of the question is incorrect there is little hope of correctly answering the question. Although question classification plays a vital role in most QA systems, many factors influence the overall ability of a system to produce the correct answer to a given question. It has been shown that parallel improvements in question classification accuracy, retrieval of candidate answer, named entity tagging, and answer extraction are needed to improve the overall performance of a QA system (Ittycheriah et al. 2000).

Online digital reference services (Pomerantz et al. to appear) represent another domain may make use of a question classification system. Here, question classification can be used as a component of a *query triage* system that determines whether a question is best answered automatically by a QA system or by a human expert based on the expected answer type. For example, a question expecting a simple result, such as a person's name, can be routed to an automatic QA system, whereas a question seeking a technical definition or a detailed explanation should likely be routed to human expert. The expected answer type may also be used by the system to choose which human expert to route the question to.

Question classification systems can be used as parts of many other applications related to information retrieval and natural language processing. This paper tries to give a domain independent overview of the subject from a machine learning perspective so as to not limit applicability to only QA systems.

### *1.2. Related work*

One of the largest QA evaluations is the *Text REtrieval Conference's* (TREC) QA track. Over the years this forum has introduced many approaches to QA and fostered a great deal of research in the field. Many of the systems use the general QA framework described above and thus make use of some form of question classification. A majority of systems use hand-crafted rules to identify expected answer types (Hull 1999, Lee et al. 1998, Prager et al. 1999). The following are examples of such rules from (Pasca and Harabagiu 2001):

What {is | are} < phrase_to_define >?
What is the definition of < phrase_to_define >?
Who {is | was | are | were} < person_name(s) >?

The first two rules detect definition questions and the last detects biographical questions. These rules have the potential to be very powerful. However, they are cumbersome to create and often do not generalize well. Hand-crafted rules that work well on a specific set of questions may give poor results when applied to another set. Rules created for a specific question ontology must be re-tailored before being applied to different ontologies. In addition to TREC QA track systems, several web-based QA systems have relied on such rules with limited success (Radev et al. 2002). Therefore, there is a need for more robust systems that can easily be adapted to handle new data sets and question ontologies.

To overcome these problems, machine learning techniques for question classification have been researched and successfully applied. Several systems make use of statistical approaches. Since it is not possible to list all such systems, we briefly describe several. Among these are IBM's TREC-9 system (Ittycheriah et al. 2000) that utilizes maximum entropy models (Della Pietra et al. 1997). It uses a mix of syntactic and semantic features (see Section 5). The authors use a data set of 1,900 questions specifically created and labeled for the task in addition to a set of 1,400 questions from a trivia database. The questions are labeled according to the MUC categories (Chinchor and Robinson 1998). On a heldout portion of the data, the system yields an accuracy of 90.95%.

Roth and Li developed a question classification system based on the Sparse Network of Winnows (SNoW) architecture (Li and Roth 2002). The system also makes use of a collection of syntactic and semantic features. The data set and question ontology they use is discussed in detail in Section 4. The system achieves 91.00% accuracy on general, coarse grained question types, and 84.20% on more specific, fine grained types.

Finally, Zhang and Lee's question classification system (Zhang and Lee 2003) is based on Support Vector Machines (SVMs) (Vapnick 1998). The system uses a tree kernel (Collins and Duffy 2002) and simple syntactic features. It is trained and tested on the same data set and question ontology used by Roth and Li. The system achieves 90.0% accuracy on the coarse grained question types.

It should also be mentioned that some work has made use of natural language processing techniques to automatically construct grammars to match question types against (Hovy et al. 2001, Nyberg et al. 2003). These systems typically make use of some underlying statistical methods, but are susceptible to poor question type coverage. The Javelin system (Nyberg et al. 2003) combines automatically learned parsers augmented with hand built rules to achieve 92.00% accuracy on a test set of TREC questions. Such parser-based approaches often perform comparably to discriminative classifiers, which is the focus of this work.

Each of these systems takes a unique statistical approach to the question classification and achieves good results (typically above 80% accuracy) on their respective data sets. Unfortunately, most past studies only present results for a single data set and provide very little in the way of error analysis. Therefore, in the remainder of this paper we explore how well statistical methods perform across several data sets. Each data set has different characteristics, such as the expressiveness of its question ontology and its source. This

allows for a broad empirical evaluation. We also examine the role different types of features have on system performance. Finally, we identify factors that hinder classification accuracy by providing an analysis and explanation of common causes of misclassification.

## 2. System overview

Before discussing the different issues involved with question classification, we first introduce the experimental framework used throughout the remainder of the paper. Like Zhang and Lee's system, our system is based on SVMs (Vapnick 1998). However, the two systems differ in a number of ways. Their system uses a single classifier, whereas we train a classifier for each unique question word. Furthermore, their system makes use of a powerful tree kernel that requires setting two parameter values, whereas we use the simpler single parameter radial basis function (RBF) kernel. Figure 1 provides a general overview of our system. The remainder of this section details how question words are extracted, feature vectors are created, SVMs are trained, and how questions are classified in our system.

### *2.1. Determining the question word*

Given a question, our system first extracts the question word. Since we only consider simple fact-based questions there is a somewhat limited lexicon of question words. Not surprisingly, the most common fact-based question words are *who*, *what*, *when*, *where*, *why*, and *how*. It is assumed that the set of question words is fixed and known *a priori*, although it can also be learned automatically. However, we use a manually generated list for simplicity. Also, some questions may not contain any of the question words in the list. A simple solution to this problem is to clump all such questions together and define their question word as *unknown*. For most fact-based questions, the question word can be extracted accurately more than 99% of the time.

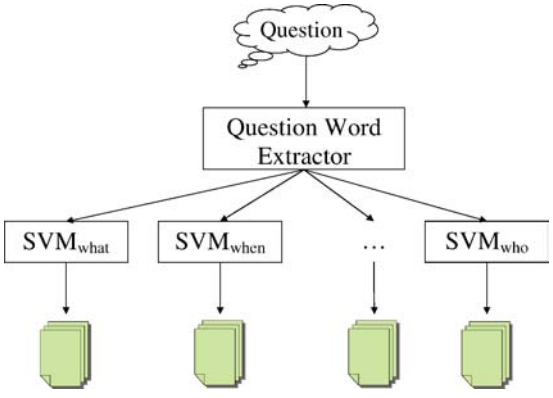


*Figure 1.* Classification system architecture.

This process partitions the data into sets, where each set corresponds to a unique question word. The data is split this way because the question word implies a great deal of information about the expected answer type. This is a form of prior knowledge that we can take advantage of. For example, questions of the form "*When is . . .* " are unlikely to expect a person's name as an answer. Instead, it is likely to be a time-related question. Unfortunately we are unable to extract this type of information from every question. Questions of the form "*What is . . .* " may be associated with many expected answer types (see Table 7). Therefore, *what*-questions provide virtually no prior information and leave a heavy burden on the shoulders of the statistical classifier.

### *2.2. Feature extraction*

After the system determines the question word it then extracts pertinent features from the question. This step is possibly the most important part of any question classification system. Better feature sets provide more accurate question representations and ultimately translate into better classification performance. The extracted features are used to create a feature vector, which is the basis for learning. Since any real-valued function from the set of possible questions to the real numbers can be a feature, there are many possibilities to choose from. However, a small set of syntactic and semantic features are most commonly used. Section 5 gives a thorough treatment of the many different kinds of features and the impact they have on system performance.

### *2.3. Learning*

The core of our statistical approach lies in the training of SVMs. We refrain from giving details of SVMs here. For a good tutorial see (Burges 1998). As figure 1 shows, rather than learn a single classifier with $k$ (=number of expected answer types) classes, our system learns $n$ (=number of distinct question words) classifiers each with $\leq k$ classes. It has been our experience that a classifier of this form typically outperforms a single monolithic classifier for this task, as it is often easier to learn several classifiers with a small number of candidate answer types than it is to learn a single classifier with many candidate answer types.

To train the SVM that corresponds to question word $q$, we use only those questions in the training set that have $q$ as their question word. Thus, each of the disjoint question sets induced as a result of identifying question words is used to train a single SVM. For example, all *who* questions in the original training set are used as training instances for the $\text{SVM}_{who}$ classifier that is depicted in figure 1. The same process is repeated for each question word.

Of course, a system could choose not to identify the question words and use a single monolithic classifier. However, there are advantages to training multiple classifiers. First, as mentioned previously, extracting the question word is a form of *a priori* information that can lead to improved performance. In essence we are minimizing the chances of a noisy classification, such as a "*Who is . . .* " question being classified as a time-related question simply because a highly "time-like" feature was extracted from the question. This is achieved at the price of less training data per classifier. Furthermore, learning multiple

classifiers allows us to train each classifier with different parameters, such as different kernels and costs (Morik et al. 1999).

### 2.4. *Classification*

After the $n$ question word specific classifiers are learned our system can be used to classify unseen questions. The classification process is simple and closely mimics the steps followed in learning. Given an unseen question, its question word is first extracted. Next, a feature vector is created using the same features used for training. Finally, the SVM corresponding to the question word is used for classification, i.e. a *what* question will be classified with $\text{SVM}_{what}$. A ranked list of expected answer types is returned based on the score generated by the SVM. Thus, the label assigned to the question is the top ranked answer type.

### 2.5. *Experimental setup*

All experiments in this paper make use of Joachims' $\text{SVM}^{light}$ (Joachims 1998) software, a one-versus-all approach to multi-class classification, RBF kernels for the SVM (Burges 1998), and 10-fold cross-validation for test set evaluation unless otherwise noted. The parameter for the RBF kernel (the variance) is set to a value that gave good performance in the past on similar classification tasks. No stopword removal or stemming is performed. Although our system makes use of SVMs, it should be noted that any multi-class statistical method can be used. Machine learning techniques applied successfully to text classification are particularly well suited for this task and include methods such as Naive Bayes (McCallum and Nigam 1998), maximum entropy models (Nigam et al. 1999), and k-nearest neighbor (Yang and Liu 1999).

This section provided a brief overview of our statistical question classification framework. Not all machine learning methods are created equal. As a result, system performance depends on the underlying learning paradigm. Regardless of this, many system-independent issues must still be resolved, such as what question ontology and set of features to use. After explaining how system performance is measured, we will explore these issues using the experimental framework developed in this section.

## 3. Performance metrics

The most common performance metric used to evaluate question classification systems is precision. Given a set of $M$ questions, their $actual$ answer types, and a ranked list of classification scores we define precision as:

$$precision = \frac{1}{M} \sum_{i=1}^{M} \delta(rank_i, 1)$$

where $\delta$ is the Kronecker delta function defined by:

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

and $rank_i$ is the rank of the correct answer type in the list returned by the classifier. Note, we assume that each question has a single correct expected answer type.

A less common, but generalized version of precision is the $P_{\leq n}$ metric. It is defined as:

$$P_{\leq n} = \frac{1}{M} \sum_{k=1}^{n} \sum_{i=1}^{M} \delta(rank_i, k)$$

The traditional definition of precision only gives credit if the correct answer type appears first in the ranked list. The generalized version is a relaxed form of this rule. It gives credit as long as the correct answer type is found anywhere in the top $n$ ranked answer types. We see that $precision = P_{\leq 1}$, $precision \leq P_{\leq n}$, and $P_{\leq n} \leq P_{\leq n+1}$ for all $n \geq 1$.

This metric provides useful information for QA systems that allow more than one expected answer type to be returned by the question classifier. For example, given the question "*Who invented the instant Polaroid camera?*", our system produces the following ranked list of question types:

| | |
|---|---|
| `person` | 0.82 |
| `organization` | −0.59 |
| `biography` | −1.05 |
| `nationality` | −1.10 |

As we see, the two top ranked expected answer types are `person` and `organization`. Rather than only retrieve passages containing `person` entity tags as possible answers the system can also include passages containing `organization` entities as potential answers as well. In this case, it may be beneficial to include both types of entities since it is not clear if the question is expecting a person or organization as an answer. We return to this kind of ambiguity again in Section 6.2.

A system that makes use of the results returned by a question classifier may wish to consider the generalized precision values. For example, in a QA system, suppose that experiments show that $P_{\leq 1} = 65\%$ and $P_{\leq 2} = 95\%$ for the question classification component. Only considering the top ranked answer type may lead to poor QA performance since there is only a 65% chance the system will extract the correct type of answer. Also considering the second ranked answer type may increase overall QA system performance, although the candidate answer list will be larger and noisier. Unfortunately there is no universal rule of thumb for choosing the best number of results to request from the classifier. A reasonable estimate can be determined by taking into account the generalized precision, system properties, and other requirements of the task.

Finally, the mean reciprocal rank (MRR), is a common metric used to evaluate QA systems that can also be used to evaluate question classification systems (Voorhees and

Tice 1999). It is calculated as:

$$\text{MRR} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{rank_i}$$

The MRR is a simple method for evaluating how well, in general, a question classification system performs. The weight of each question's classification is inversely proportional to how well the question was classified. If a system achieves *precision* $= 100\%$, then the MRR $= 1.0$ since $rank_i = 1$ for all $i$. Furthermore, if a system assigns the correct answer type the highest score in the ranked list half of the time and assigns it the second highest score the other half of the time, then MRR $= \frac{1}{M}(\frac{M}{2} \cdot 1 + \frac{M}{2} \cdot \frac{1}{2}) = \frac{3}{4}$. We will primarily use MRR to evaluate the overall effect different feature sets have on our system.

## 4. Data

There are many potential data sources for question classification. Here, a data set is defined as a collection of questions labeled with expected answer types drawn from some question ontology. The ontology is not specific to the questions and is typically chosen to meet the task requirements. To validate how robust a question classification system is, experiments must be done on a number of diverse data sets. We explore three data sets with varying qualities. Each differs in size, source, question ontology, and underlying style. These data sets are used throughout the remainder of the paper to empirically explore how different factors affect classification accuracy.

### *4.1. TREC QA track questions*

As discussed in Section 1, the TREC QA track is a large scale QA evaluation first introduced at TREC-8 (Voorhees and Tice 1999). Each year a new set of questions is created for the track. The questions do not come labeled with an expected answer type. Many different groups participating in TREC have created their own question ontologies and used them to label the TREC questions. There are no fixed guidelines for creating a question ontology. However, since most systems perform named entity tagging on retrieved passages to find candidate answers, it is likely that a system's question ontology contains similar types to those that the named entity tagger is capable of extracting. Since BBN's IdentiFinder (Bikel et al. 1999) named entity tagger is widely used, we chose to use BBN's question ontology, which consists of 31 answer types. BBN also provided us with labeled TREC-8, 9, and 10 QA track questions.

Each year the set of questions are drawn from a different source and generally have different characteristics. The 200 TREC-8 questions were specifically created for the task. The TREC-9 questions were extracted from Encarta and Excite logs. It consists of 500 original questions and 193 additional questions that are variants on the original set. Variants of 54 original questions were included. An example of an original question and its variants is Voorhees (2000):

Original: *What is the tallest mountain?*
Variants: *What is the world's highest peak?*
*What is the highest mountain in the world?*
*Name the highest mountain.*
*What is the name of the tallest mountain in the world?*

Finally, the TREC-10 questions were drawn from MSNSearch and AskJeeves search logs (Voorhees 2001). The 1,393 TREC-8,9, and 10 questions combined form what we refer to as the TREC QA data set. Although these represent real question, it should be noted that NIST accessors do their best to correct spelling, punctuation, and grammatical errors. Despite their efforts, several questions still contain such errors. Even with these errors, the data set is far from realistic. The MadSci questions described in Section 4.3 are a more realistic set of questions.

Table 1 lists the 31 BBN question types and gives an example of each. A question is assigned the label `other` if it does not fit into one of the other 30 categories. Figure 2 shows the distribution of each question type within the data set. The distribution is very skewed. The four question types `date`, `definition`, `gpe` (geo-political entity), and `person` account for approximately 57% of all questions. At the opposite end of the spectrum, `facility` `description` and `time` each correspond to only a single question in the data set. Therefore, for most statistical machine learning techniques, the less commonly occurring question types are difficult to classify correctly due to this data sparsity.

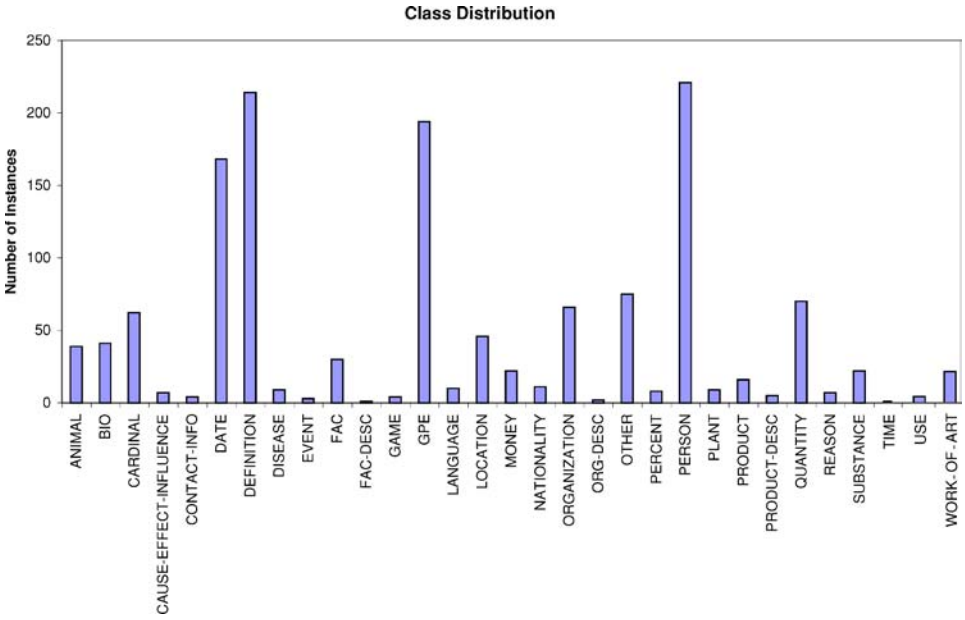


*Figure 2.* Number of questions per question type for the TREC QA track data set using the BBN question ontology.

*Table 1*. BBN question ontology and a sample question for each question type.

| Example question | Question type |
|---|---|
| What do you call a group of geese? | Animal |
| Who was Monet? | Biography |
| How many types of lemurs are there? | Cardinal |
| What is the effect of acid rain? | Cause/Effect/Influence |
| What is the street address of the White House? | Contact Info |
| Boxing Day is celebrated on what day? | Date |
| What is sake? | Definition |
| What is another name for nearsightedness? | Disease |
| What was the famous battle in 1836 between Texas and Mexico? | Event |
| What is the tallest building in Japan? | Facility |
| What type of bridge is the Golden Gate Bridge? | Facility description |
| What is the most popular sport in Japan? | Game |
| What is the capital of Sri Lanka? | Geo-political entity |
| Name a Gaelic language. | Language |
| What is the world's highest peak? | Location |
| How much money does the Sultan of Brunei have? | Money |
| Jackson Pollock is of what nationality? | Nationality |
| Who manufactures Magic Chef appliances? | Organization |
| What kind of sports team is the Buffalo Sabres? | Organization description |
| What color is yak milk? | Other |
| How much of an apple is water? | Percent |
| Who was the first Russian astronaut to walk in space? | Person |
| What is Australia's national flower? | Plant |
| What is the most heavily caffeinated soft drink? | Product |
| What does the Peugeot company manufacture? | Product description |
| How far away is the moon? | Quantity |
| Why can't ostriches fly? | Reason |
| What metal has the highest melting point? | Substance |
| What time of day did Emperor Hirohito die? | Time |
| What does your spleen do? | Use |
| What is the best-selling book of all time? | Work of art |

As a baseline for comparing the effect of different data sets and feature types, we present the results of our classification system for this data set using *bag of words* features. That is, the features extracted from each question consist only of the individual words that make up the question. This is one of the simplest feature representations. Table 2 summarizes the results using the $P_{\leq n}$ metric for several values of $n$ and the *MRR*.

*Table 2.* TREC QA track results using simple word features.

| $n$ | $P_{\leq n}$ |
|---|---|
| 1 | 77.59 |
| 2 | 86.42 |
| 3 | 89.51 |
| 4 | 91.16 |
| 5 | 92.60 |
| 10 | 95.33 |
| MRR | 0.8437 |

### *4.2. UIUC questions*

In Li and Roth (2002), Li and Roth use a superset of the TREC QA track questions and impose a different question ontology on the data. This is what we refer to as the UIUC data set. The *training* data they use consists of the TREC-8 and 9 QA track questions, 4,500 questions from a USC data set, and approximately 500 manually constructed questions to cover rare question types (Li and Roth 2002). Additionally, for *testing* purposes they use the 500 TREC 10 QA track questions.[1] As with the TREC QA questions, these questions have proper grammar and spelling and again are rather ideal.

What makes this data set distinct from the TREC QA track questions is the question ontology. Rather than using a *flat* ontology they make use of a *hierarchical* one. The hierarchy, shown in Table 3, has 6 coarse grained classes and 50 fine grained classes. Such a hierarchical ontology allows more flexibility than the flat one discussed previously. It allows us to classify questions at varying degrees of granularity and possibly take advantage of the hierarchical nature when learning. Systems using the output of the classifier may also be able to make use of the hierarchy in different ways.

Again, we provide baseline results for our system on the UIUC data set. Note, unlike experiments done on the other two data sets that use 10-fold cross validation, all experiment on this data set throughout the paper use the 5,500 questions discussed above for training and the 500 TREC-10 questions for testing. Table 4 gives system performance results for both coarse and fine grained question types.

Caution should be taken when considering the coarse grained question type results. Since there are only 6 question types, $P_{\leq 10}$ is trivially 100%. Also, if the question types were *randomly* ranked, then $P_{\leq 5} = \frac{5}{6}$ (83%). More generally, given $T$ different question types, a random ranking of question types yields $P_{\leq n} = \frac{n}{T}$ for $n = 1 \ldots T$ and $P_{\leq n} = 1$ (100%) for $n \geq T$.

As Table 4 shows, our system performs better on the coarse grained classes. The results expose the tradeoff between ontology specificity and accuracy. This can be attributed to the fact that the performance of statistical machine learning techniques depends on the amount of quality training data. Since there are 6 coarse grained and 50 fine grained question types, each of the coarse types contain significantly more training data per class than the fine

*Table 3.* UIUC hierarchical ontology. Coarse grained question types are italicized. For example, if a question is classified as the coarse grained type `human` it is then one of the following fine grained types: `human:group`, `human:individual`, `human:title`, or `human:description`.

| Coarse | Fine |
|---|---|
| *abbreviation* | abbreviation, explanation |
| *entity* | animal, body, color, creative, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word |
| *description* | definition, description, manner, reason |
| *human* | group, individual, title, description |
| *location* | city, country, mountain, other, state |
| *numeric* | code, count, date, distance, money, order, other, period, percent, speed, temperature, size, weight |

*Table 4.* UIUC results using simple word features.

| $n$ | Coarse $P_{\leq n}$ | Fine $P_{\leq n}$ |
|---|---|---|
| 1 | 86.20 | 81.00 |
| 2 | 95.60 | 87.20 |
| 3 | 98.80 | 90.20 |
| 4 | 99.80 | 92.00 |
| 5 | 100.0 | 93.40 |
| 10 | 100.0 | 95.60 |
| MRR | 0.9224 | 0.8628 |

grained types. As mentioned above, approximately 500 of the questions in the data set were specifically created to overcome this data sparsity problem. This allows the classifier to achieve reasonable performance using the fine grained question types. However, in a real setting, such as the MadSci data we discuss next, such nicely distributed data may not be available.

### *4.3. MadSci questions*

The MadSci data set consists of science related fact-based questions culled from the MadSci[2] question archive. MadSci is a science web page that provides a way for users of all ages and backgrounds to ask scientific questions and receive answers from experts. To ask a question a user inputs their grade level, the area of science the question relates

to, their actual question, and any optional comments or further information they choose to include. The only information we consider is the text of the question. The other information can be used to further enhance question classification, but is not used at present.

The entire MadSci archive consists of 12,348 questions. From this collection we randomly sampled 250 questions. A highly accurate classifier based on regular expression matching was used to discriminate between fact-based and task-oriented questions to ensure only fact-based questions were included in the sample (Murdock and Croft 2002). Throughout the remainder of this paper these 250 questions will be referred to as the MadSci data set.

The 250 questions were labeled by hand using an augmented version of the TREC QA ontology. We found that many questions could not be labeled under the original ontology since the questions were inherently different. Thus, two new question types were added to avoid a large percentage of questions being labeled `other`. The first new type, `choose-list`, is for questions seeking an answer from a list, such as: "*Were dinosaurs coldblooded or warmblooded animals?*". Second, the question type `yes-no-explain` was added for questions that expected a yes or no answer and an optional explanation, such as: "*Can scientist create atoms or is it impossible to be manmade?*"

As the two examples in the previous paragraph illustrate, the MadSci questions are not as ideal as the questions in the two other data sets. This is the most realistic of the three data sets and presents more of a challenge to our classifier. The following are more unmodified examples taken from the data set:

*does time go frame by frame like in a movie or is it an endless continuum?*
*what is the h323 protocol and t30 protocol?*
*can i turn my ceilling fan into a neg. ion generator by using teflon blades?*
*which is hotter the sun or lightening?*

Table 5 shows the baseline system performance for the MadSci data set using bag of words features. This set of questions exhibits the worst performance of the three both as a result of the realistic, noisy data and small data set size.

*Table 5.* MadSci results using simple word features.

| $n$ | $P_{\leq n}$ |
|---|---|
| 1 | 72.60 |
| 2 | 83.60 |
| 3 | 86.80 |
| 4 | 90.00 |
| 5 | 90.80 |
| 10 | 94.40 |
| MRR | 0.8124 |

### *4.4. Discussion*

As the baseline results for the three data sets show, our statistical approach to question classification is robust. Each data set discussed has very different qualities, yet our system was able to achieve relatively good performance on each using simple features. From the results we see that statistical classifiers can be robust across multiple data sets with varying characteristics. The same generally cannot be said of traditional hand-crafted rules. For each data set a different collection of rules would have to be manually created, which is a timely, expensive process. For supervised statistical classifiers, such as the one presented here, only a set of labeled questions is necessary. Creating such a set is often less time consuming and requires less domain knowledge than hand crafting rules. Furthermore, it is often possible to obtain labeled training data without any human intervention (Davidov et al. 2004). Thus, statistical classifiers, such as our SVM-based system, may be applied to a wide range of data sets and question ontologies with very little or no hand tuning and manual effort necessary.

From these results we also see that a great deal of information can be learned by looking at more than just the answer type associated with the highest classification score. For example, for the TREC QA questions, only considering the top answer type results in an accuracy of 77.59%. However, if the second most likely answer type is also considered the accuracy jumps to 86.42%. As discussed in Section 3, the $P_{\leq n}$ metric should be considered along with other system properties and requirements to determine how to best use the question type list returned by the question classifier. In the case of the TREC QA questions, it may be beneficial to make use of the top two types returned by the classifier.

## 5. Features

We showed in the previous section that a bag of words feature representation results in relatively good performance across all three data sets. In this section we explore syntactic and semantic question features, and empirically evaluate the impact these richer feature sets have on system performance.

### *5.1. Syntactic features*

Syntactic features are used to represent or encode the syntax of a question. They are appealing because questions of the same type often have the same syntactic style. That is, they often share a similar structure and vocabulary. The simplest syntactic features are $k$-grams. A $k$-gram is an ordered arrangement of $k$ words. For $k = 1$ such features are called *unigrams* and for $k = 2$ they are called *bigrams*. The bag of words features discussed previously are simply unigrams. Higher order $k$-grams allow us to exploit dependencies between words. For example, consider a question beginning with "*How far...*". Unigram features are incapable of explicitly expressing that *how* is followed by *far*. However, bigram features allow us to explicitly model the dependence of these two words. Using bigrams rather than unigrams could allow us to learn that a sentence containing the phrase *how far* is likely a quantity related question. However, using higher order $k$-grams causes the number of features to

*Table 6.* Results using bigram features.

| $n$ | TREC QA $P_{\leq n}$ | UIUC coarse $P_{\leq n}$ | UIUC fine $P_{\leq n}$ | MadSci $P_{\leq n}$ |
|---|---|---|---|---|
| 1 | 78.81 (+1.57)† | 87.20 (+1.16) | 81.20 (+0.25) | 73.20 (+0.83) |
| 2 | 86.85 (+0.50) | 95.80 (+0.21) | 87.60 (+0.47) | 82.80 (−0.96) |
| 3 | 90.30 (+0.88)† | 99.00 (+0.20) | 89.80 (−0.43) | 87.60 (+0.92) |
| 4 | 91.74 (+0.64)‡ | 99.80 (+0.00) | 91.60 (−0.43) | 90.40 (+0.44)‡ |
| 5 | 92.82 (+0.24) | 100.0 (+0.00) | 92.40 (−1.07) | 92.00 (+1.33)† |
| 10 | 95.40 (+0.07) | 100.0 (+0.00) | 94.60 (−1.05) | 94.40 (+0.00) |
| MRR | 0.8517 (+0.95) | 0.9279 (+0.60) | 0.8623 (−0.06) | 0.8111 (−0.16) |

explode and the amount of data for each feature to become sparse. Therefore systems rarely use anything more than unigrams and bigrams. Table 6 shows system performance on the three data sets using both unigram and bigram features. The values in parenthesis represent the relative (percentage) difference in performance compared to the unigram baseline results. Furthermore, results that are statistically significant ($p < 0.05$) over the baseline are denoted by † and those that are weakly significant ($p < 0.10$) are denoted by ‡, as determined by a signed t-test (Yang and Liu 1999). We see that adding bigram features almost always increases precision marginally, although only significantly for the TREC QA track questions. Interestingly, bigram features cause the MRR to decrease for both the fine grained UIUC questions and the MadSci questions. Such behavior is the result of data sparsity. Very few bigrams appear more than once in each of the 50 fine grained categories and in the small MadSci sample, whereby the bigram features add unnecessary complexity and little in the way of information.

Part of speech (POS) tags provide another set of syntactic features. Our system uses the maximum entropy model based MXPost (Ratnaparkhi 1996) for POS tagging. Unfortunately, incorporating POS tags explicitly as features in our system fails to yield improved accuracy. As discussed in Section 1.2, SVMs allow POS tags to be used implicitly via a tree kernel. POS tags used in this way have been shown to improve system performance (Zhang and Lee 2003).

There exists a multitude of other syntactic features. One possible feature for question classification is the question word. Such a feature is irrelevant for our system, since we use question words explicitly to partition the data. However, other systems may make use of it. Although the question word is included implicitly as a feature when using unigrams, it can potentially be beneficial to include it as a separate feature. This can help classify questions such as: "*What color does litmus paper turn when it comes into contact with a strong acid?*". Here, both *what* and *when* appear in the question, but *what* is the actual question word. Knowing it is a *what* question avoids automatically misclassifying the question as a time or date-related question.

Another possible feature is presence of a proper noun phrase. Questions with proper noun phrases, such as names or places often are questions about locations or asking for

biographical information. Other possible features include question length, noun phrases, and long distance $k$-grams (Rosenfeld 1996). None of these additional syntactic features have showed significant performance improvements when used in our system.

### 5.2. *Semantic features*

It is possible to achieve reasonable results using syntactic features alone. However, some questions, such as *what* questions, are often incorrectly classified when only syntactic features are used. Table 7 shows a sample of *what* questions from the TREC QA track data. These questions comprise 23 different question types. Knowing that a question begins with *what* provides little information about the question type. Other syntactic features of the sentence also reveal little. The words italicized in Table 7 are those words that provide clues as to the correct question type. For example, for the question "*What is the tallest mountain?*", knowing that a mountain is a location allows us to assign the correct type. Notice that the syntactic features of this sentence provide very little information. Knowing the sentence contains the word *mountain* is not enough to correctly classify the question, because *mountain* may not appear in any other `location` questions. This question is nearly syntactically identical to "*What is the tallest building in Japan?*", a `facility` question. Being able to differentiate between the meaning of *mountain* and *building* is the key factor in correctly classifying these questions.

Therefore we need a way to include semantic features to solve some of these problems. A powerful natural language processing and linguistic tool is WordNet (Fellbaum 2000). WordNet is a lexical database that provides a wealth of semantic information. A heuristic, yet simple way to incorporate WordNet features is to to extract semantic information about the headword of the main noun phrase for each question. The main noun phrase of a sentence contains the focus of the sentence, and the headword can be thought of as the "important" noun within the phrase. For example, for the question "*What is Nicholas Cage's profession*" the main noun phrase is *Nicholas Cage's profession* and the headword is *profession*. Returning to the *tallest building* and *tallest mountain* example, the headwords are *building* and *mountain*, respectively. These are precisely the words we identified as being important discriminators. To extract headwords we apply a simple heuristic. First, we run a POS tagger on each question. Next, we find the first noun phrase based on the POS tags and assume it is the main noun phrase. Finally, ignoring post-modifiers such as prepositional phrases, we extract the rightmost word tagged as a noun. We then extract this term as the headword. Although this method is heuristic and highly sensitive to the POS tagger output, it accurately finds the headwords approximately 90% of the time on the TREC questions.

Next, we use WordNet to determine the hypernyms of the headword. Hypernyms can be thought of as semantic abstractions. For instance, some of the hypernyms for *dog* are: *canine*, *carnivore*, *mammal*, *animal*, and *living thing*. Therefore, hypernyms capture a great deal of semantic information about the word and can be used to overcome some limitations brought about by using purely syntactic features. All of the hypernyms of the headword returned by WordNet are included as features. There has been work on automatically choosing the best hypernym to use to describe a given term (Prager et al. 2001). However, such an approach is not necessary here. The statistical classifier will determine which hypernyms are the most

*Table 7.* *What* questions from the TREC QA track data set representing 23 different question types. The primary words that humans use the meaning of to classify the question correctly are italicized.

| Question | Type |
|---|---|
| What is the proper name for a female *walrus*? | `animal` |
| What is Nicholas Cage's *profession*? | `bio` |
| What is the *population* of Seattle? | `cardinal` |
| What *caused* the Lynmouth floods? | `cause-effect-influence` |
| What is the *telephone number* for the University of Kentucky? | `contact info` |
| What *time of year* do most people fly? | `date` |
| What is the name of the art of growing miniature trees? | `definition` |
| What is another name for *nearsightedness*? | `disease` |
| What was the name of the famous *battle* in 1836 between Texas and Mexico? | `event` |
| What is the tallest *building* in Japan? | `facility` |
| What was the most popular *toy* in 1957? | `game` |
| What is the *capital* of Uruguay? | geo-political `entity` |
| What *language* is mostly spoken in Brazil? | `language` |
| What is the tallest *mountain*? | `location` |
| What *debts* did Qintex group leave? | `money` |
| What is the *cultural origin* of the ceremony of potlatch? | `nationality` |
| What is the name of the chocolate *company* in San Francisco? | `organization` |
| What is done with worn or outdated flags? | `other` |
| What is the name of Neil Armstrong's *wife*? | `person` |
| What is the most heavily caffeinated *soft drink*? | `product` |
| What is the average *weight* of a Yellow Labrador? | `quantity` |
| What *metal* has the highest melting point? | `substance` |
| What did Shostakovich *write* for Rostropovich? | `work of art` |

discriminative for a given question type during training. This essentially chooses the best hypernyms automatically.

Table 8 shows the results and comparison to the baseline of unigram only features when we add WordNet hypernyms to the feature representation of *what*, *which* and *name* questions. As the table shows, WordNet features lead to improvements for nearly every performance measure, even the MadSci data set that contains many grammatically irregular sentences. Also, the TREC QA track question's MRR is increased significantly over the baseline. The table provides evidence that semantic features can increase performance more than simple syntactic features such as bigrams.

Li and Roth make use of a number of semantic features in Li and Roth (2002). Their use of "related words" is similar in nature to the method just described. Rather than automatically extracting a word and expanding it using WordNet, though, they manually create a list of

*Table 8.* Results using WordNet features.

| $n$ | TREC QA $P_{\leq n}$ | UIUC coarse $P_{\leq n}$ | UIUC fine $P_{\leq n}$ | MadSci $P_{\leq n}$ |
|---|---|---|---|---|
| 1 | 80.39 (+3.61)$^{\dagger}$ | 88.20 (+2.33)$^{\ddagger}$ | 82.20 (+1.48) | 74.80 (+3.05) |
| 2 | 90.45 (+4.66)$^{\dagger}$ | 96.80 (+1.26)$^{\ddagger}$ | 89.60 (+2.75)$^{\dagger}$ | 84.40 (+0.97) |
| 3 | 93.10 (+4.01)$^{\dagger}$ | 98.80 (+0.00) | 92.40 (+2.44)$^{\dagger}$ | 88.80 (+2.32)$^{\dagger}$ |
| 4 | 95.55 (+4.82)$^{\dagger}$ | 99.80 (+0.00) | 93.40 (+1.52)$^{\ddagger}$ | 91.60 (+1.79)$^{\dagger}$ |
| 5 | 97.34 (+5.12)$^{\dagger}$ | 100.0 (+0.00) | 94.20 (+0.86) | 93.60 (+3.10)$^{\dagger}$ |
| 10 | 95.40 (+0.07)$^{\dagger}$ | 100.0 (+0.00) | 96.40 (+0.84) | 95.60 (+1.27)$^{\dagger}$ |
| MRR | 0.8727 (+3.44)$^{\dagger}$ | 0.9344 (+1.30) | 0.8768 (+1.62)$^{\ddagger}$ | 0.8250 (+1.55) |

semantically related words for each question type. If a word that appears in a question is also in one of these lists then a feature is set indicating the sentence contains a word that is often related to some question type. Although this method is effective, the list of related words for each question type must be created by hand, whereas the WordNet method discussed here is automatic.

Their system also extracted named entities from the questions. Such features also capture semantic information. In their system, this led to an improvement in performance. However, we found this actually degrades performance with our system because the entity types that appear in a question often do not correlate with the question types.

The use of syntactic features for question classification is well studied. However, the space of semantic features remains largely unexplored beyond the use of WordNet and named entity tagging (Nyberg et al. 2003). Based on the results presented here and in other works that make minimal use of semantic features it seems fruitful to explore this direction more in the future.

### *5.3. Discussion*

Finally, we present results from combining a number of features discussed above. Table 9 shows results for our system using the following combination of features: unigrams, bigrams, WordNet hypernyms, and proper noun phrase presence.

The combined feature set outperforms the baseline results in terms of *MRR* and $P_{\leq 1}$. For all data sets, excluding MadSci, precision is increased significantly. Furthermore, for both the TREC QA track and UIUC fine grained questions the MRR is increased significantly. This feature set represents the best performance achieved by our system on the TREC QA and UIUC coarse and fine grained question sets. However, for the MadSci data, using unigram and WordNet features yielded the best results. Such poor generalization is largely due to the small training set and large number of features. Thus, for larger data sets combining both syntactic and semantic features can lead to significantly better system performance.

*Table 9.* Results using bigram and WordNet features.

| $n$ | TREC QA $P_{\leq n}$ | UIUC coarse $P_{\leq n}$ | UIUC fine $P_{\leq n}$ | MadSci $P_{\leq n}$ |
|---|---|---|---|---|
| 1 | 81.25 (+4.72)$^{\dagger}$ | 90.20 (+4.65)$^{\dagger}$ | 83.60 (+3.22)$^{\dagger}$ | 73.20 (+0.83) |
| 2 | 88.58 (+2.50)$^{\dagger}$ | 95.00 (−0.63) | 88.00 (+0.93) | 82.80 (−0.96) |
| 3 | 93.32 (+4.26)$^{\dagger}$ | 97.80 (−1.01) | 90.40 (+0.22) | 88.00 (+1.38)$^{\ddagger}$ |
| 4 | 94.40 (+3.55)$^{\dagger}$ | 99.60 (−0.20) | 91.80 (−0.20) | 90.80 (+0.44) |
| 5 | 95.33 (+2.95)$^{\dagger}$ | 100.0 (+0.00) | 93.40 (+0.00) | 93.60 (+0.89)$^{\dagger}$ |
| 10 | 97.20 (+1.96)$^{\dagger}$ | 100.0 (+0.00) | 95.80 (+0.21) | 95.60 (+1.27)$^{\dagger}$ |
| MRR | 0.8737 (+3.56)$^{\dagger}$ | 0.9405 (+1.96)$^{\ddagger}$ | 0.8771 (+1.66)$^{\dagger}$ | 0.8134 (+0.12) |

## 6. Error analysis

In this section we explore common causes of classification error to develop a better understanding of the limiting factors involved with statistical question classification. We explore issues involving data labels, question difficulty, POS tagger errors, and WordNet insufficiencies. Throughout this section we primarily focus is on the TREC QA track data set. However, all analysis provided is general and valid for other data sets as well.

### *6.1. Inconsistent and ambiguous labeled data*

With any statistical method that learns on training data, the resulting classifier is only as good as the data that is given to it. An analysis of the incorrectly classified questions revealed that a number of the errors were the result of incorrectly labeled data. The following question/answer type pairs taken from the TREC data set illustrate the point:

*Who is Duke Ellington?* `person`
*Who is Charles Lindbergh?* `biography`

*What does CNN stand for?* `organization`
*What does USPS stand for?* `definition`

Clearly each pair of questions should have the same data label since they are both requesting the same type of answer. Hand labeling data is a monotonous human task and thus doomed to contain errors. However, the fact that a question may not cleanly fit into a single question type only compounds the problem. A number of questions have ambiguous classifications. The following questions are labeled as `facility`, but are equally valid as `location` or `gpe` questions depending on the exact information need of the user:

*Where is the actress, Marion Davies, buried?*
*Where was Lincoln assassinated?*

Both of these questions are classified as `gpe` by our system. Such a classification is not necessarily incorrect. Such ambiguity arises from the inability of the question ontology to properly assign a single question type to the question. This problem can be overcome by allowing each question to have multiple labels or by making use of a different question ontology (Hovy et al. 2001, 2002).

For the TREC QA track data set, the most commonly misclassified questions, above and beyond `other` questions are those that belong to closely related question types, such as {`gpe, location`}, {`quantity, cardinal, percent`, and `money`}, and {`person` and `organization`}. Combining these pairs of classes would likely result in better performance at the expense of less specific question classification. This is the strategy employed with the coarse versus fine grained UIUC questions. Based on the results presented in Section 5, we see that classification precision improves as ontology generality increases. However, no single question ontology is the best choice for all tasks. Instead, an ontology should be chosen based on the task and other characteristics of the system keeping in mind the tradeoff between classification accuracy and answer type specificity.

### *6.2. Inherently difficult questions*

Next, there are some questions that are inherently ambiguous and/or difficult to classify. These questions ultimately require hand built rules, a deep human understanding, or advanced natural language processing techniques to be classified correctly. Some examples, again taken from the TREC data set, are:

*What is the name of the Lion King's son in the movie "The Lion King"?* `animal`

*Who developed potlatch?* `nationality`

*Name the designer of the shoe that spawned millions of plastic imitations, known as "jellies".* `organization`

Each of these questions require a more general knowledge of the world or context to accurately predict the question type. For instance, in the first example above it is crucial to know that the Lion King is an animal, otherwise there is little hope of knowing the expected answer type is *animal*. In the second example, a working knowledge of Native American ceremonies is required to correctly classify the question type. Finally, in the last example, it it not clear whether the question expects the name of the person who designed the shoe or the name of the organization that designed the shoe. This issue is closely related to our discussion of ambiguous question types.

### *6.3. POS tagger and WordNet expansion error*

Although POS taggers are capable achieving high accuracy, they are not infallible. POS tags are necessary in the method described in Section 5.2 to determine which word(s) to expand using WordNet. Thus, an error in the POS tagger will be propagated through to WordNet

expansion and may ultimately affect the classification. This can cause the classifier to believe a given *what* question pertains to animals rather than a location. Fortunately, these problems do not seem to hurt performance significantly since the SVMs can overcome most errors introduced this way. However, it often leads to easy questions being misclassified, such as the following question:

Question: *What U.S. Government agency registers trademarks?* `organization`
Tagged: *What_WP U.S._NNP Government_NN agency_NN registers_NNS trademarks_NNS ?_.*

The tagger incorrectly tags *registers* as a plural noun. Using the heuristic headword extractor described previously results in *trademarks* being expanded via WordNet. Ultimately this causes the question to be incorrectly classified. If the correct headword (*agency*) were expanded instead then the question most likely would have been correctly classified as *organization*. Possible ways to overcome POS tag error and expanding the incorrect word is to use a more accurate POS tagger or a less heuristic method of extracting the headword, such as producing a full parse tree of the question.

### *6.4. WordNet insufficiencies*

Although WordNet is an excellent natural language processing and linguistic tool, we encounter some problems when trying to use it for question classification. WordNet is primarily used to help classify *what* questions. For these questions we assume that the headword provides the most evidence about the expected answer type. This assumption holds for a majority of questions. However, we must also assume that WordNet provides a good abstraction for the headword via its hypernym hierarchy. For example, an ideal abstraction for the words *cat*, *dog*, and *walrus* is *animal*. Unfortunately, it is not that simple. WordNet's hypernym hierarchy is very complex. Problems arise when it fails to express the most basic human understanding of a word or when it fails to reveal a connection between strongly related words. For example, consider the following question:

Question: *What cereal goes "snap, crackle, pop"?* `product`
Tagged: *What_WP cereal_NN goes_VBZ "_" snap_NN ,_, crackle_NN ,_, pop_NN "_" ?_.*

The headword of this question is *cereal*. When most humans think of *cereal*, they first think of breakfast food. To humans, it very obvious that *cereal* is a breakfast food, and that foods are products. However, WordNet returns three senses of the word cereal. If we assume that users asking questions use simple vocabulary, then we can assume they use the most common or basic meaning of words. For *cereal*, the first sense is "*cereal, cereal grass*", which concerns plants. This expansion is detrimental to classifying this question as *product*. The second and third senses of *cereal* are more familiar senses of the word. Therefore, the ordering of the hypernyms isn't always intuitive. Although WordNet expansion introduces a certain amount of noise into our data, it does improve classification as we showed in Section 5.2. The use

of word sense disambiguation techniques (Brown et al. 1991) could be useful to overcome such issues. Furthermore, other semantic ontologies such as the Suggested Upper Merged Ontology (SUMO) (Niles and Pease 2001) have been created based on WordNet and may be provide useful tools for extracting better semantic meaning from questions. Clearly, if a system used a perfect POS tagger, a perfect method of extracting headwords, and a perfect way of abstracting a noun to a general idea, then *what* questions could be classified with very high accuracy.

## 7. Conclusion

In this paper, we presented an overview of statistical question classification applied to fact-based questions. Many past approaches resorted to building specialized hand-crafted rules for each question type. Although such rules prove effective, they do not scale well and are tedious to create. Statistical classifiers provide a more robust framework for exploring question classification. Our statistical classifier is based on SVMs and uses prior knowledge about correlations between question words and types to learn question word specific classifiers. Under such a statistical framework, any data set, question ontology, or set of features can be used.

We showed empirically that statistical classifiers are robust in handling different types of features. In general, semantic features are more powerful than syntactic features. They endow the statistical classifier with a certain understanding of a question's meaning via the use of WordNet hypernyms. Furthermore, combining both syntactic and semantic features allows for the most flexibility and generally achieves better performance, increasing precision significantly on three of the data sets.

The analysis of common misclassifications gives insight into possible improvements to our system and other statistical classifiers. First, data sets require expressive yet unambiguous question ontologies to guard against mislabeling errors or ambiguities. Next, more sophisticated labeling methods, such as allowing questions to be associated with one or more question types, are necessary to overcome the problem of questions with ambiguous question types. Although this may solve some problems, difficult questions will always exist that escape being correctly classified with only a limited understanding of the question. Also, inaccurate POS taggers often cause errors to propagate through to the final result. Finally, WordNet's ability to abstract common concepts can lead to problems when it is used to extract features in a system, as we showed in the case of *cereal*. Therefore, natural language processing techniques, such as word sense disambiguation, could prove to be beneficial.

There are many avenues of future work left to explore. Our results show that simple semantic features can improve system performance more than syntactic features. Unfortunately, these features are not as well studied or understood compared to their syntactic counterparts. Therefore, more advanced methods of including semantic information need to be explored. Also, better question ontologies need to be developed to be both expressive enough to cover most question types and as unambiguous as possible. If the ontology is being designed for a QA system, we must also take into account whether or not a named entity tagger can extract candidate answers for each question type. Finally, a larger realistic

data set like the MadSci data should be created as a standard test collection that would allow comparison across different classification systems and help further advance the state of the art.

## Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant number DUE-0226144 and in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

## Notes

1. The UIUC data set is available at http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC.
2. http://www.madsci.org

## References

Bikel DM, Schwartz RL and Weischedel RM (1999) An algorithm that learns what's in a name. Machine Learning, 34(1–3):211–231.

Brown PF, Pietra SD, Pietra VJD and Mercer RL (1991) Word-sense disambiguation using statistical methods. In: Meeting of the Association for Computational Linguistics, pp. 264–270.

Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121–167.

Chinchor N and Robinson P (1998) MUC-7 named entity task definition. In: Proceedings of MUC-7.

Collins M and Duffy N (2002) Convolution kernels for natural language. In: Dietterich TG, Becker S and Ghahramani Z, eds., Advances in Neural Information Processing Systems 14. Cambridge, MA.

Davidov D, Gabrilovich E and Markovitch S (2004) Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In: Proceedings of the 27th annual international conference on Research and development in information retrieval, pp. 250–257.

Della Pietra S, Della Pietra VJ and Lafferty JD (1997) Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):380–393.

Fellbaum C, ed. (2000) WordNet: An Electronic Lexical Database. MIT Press.

Hovy E, Gerber L, Hermjakob U, Lin C-Y and Ravichandran D (2001) Towards semantics-based answer pinpointing. In: Proceedings of the DARPA Human Language Technology Conference (HLT).

Hovy E, Hermjakob U and Ravichandran D (2002) A question/answer typology with surface text patterns. In: Human Language Technology Conference (HLTC).

Hull D (1999) Xerox TREC-8 question answering track report. In: Proceedings of the 8th Text Retrieval Conference (TREC-8).

Ittycheriah A, Franz M, Zhu W and Ratnaparkhi A (2000) IBM's statistical question answering system. In: Proceedings of the 9th Text Retrieval Conference (TREC-9).

Joachims T (1998) Making large-scale support vector machine learning practical. In: Schölkopf ASB, Burges C, eds., Advances in Kernel Methods: Support Vector Machines.

Lee K-S, Oh J-H, Huang J, Kim J-H and Choi K-S (2003) TREC-9 Experiments at KAIST: QA, CLIR and batch filtering. In: Proceedings of the 9th Text Retrieval Conference (TREC-9).

Li X and Roth D (2002) Learning question classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics.

McCallum A and Nigam K (1998) A comparison of event models for naive bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization.

Morik K, Brockhausen P and Joachims T (1999) Combining statistical learning with a knowledge-based approach—a case study in intensive care monitoring. In: Proceedings of 16th International Conference on Machine Learning, pp. 268–277.

Murdock V and Croft WB (2002) Task orientation in question answering. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 355–356.

Nigam K, Lafferty J and McCallum A (1999) Using maximum entropy for text classification. In: Proceedings of Machine Learning for Information Filtering Workshop IJCAI'99, pp. 61–67.

Niles I and Pease A (2001) Towards a standard upper ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems, pp. 2–9.

Nyberg E, Mitamura T, Callan J, Carbonell J, Frederking R, Collins-Thompson K, Hiyakumoto L, Huang Y, Huttenhower C, Judy S, Ko J, Kupsc A, Lita LV, Pedro V, Svoboda D and Durme BV (2003) The JAVELIN question-answering system at TREC 2003: A multi-strategy approach with dynamic planning. In: Proceedings of the 12th Text Retrieval Conference (TREC 2003).

Pasca M and Harabagiu SM (2001) High performance question/answering. In: Research and Development in Information Retrieval, pp. 366–374.

Pomerantz J, Nicholson S, Belanger Y and Lankes RD (to appear) The current state of digital reference: Validation of a general digital reference model through a survey of digital reference services. Information Processing and Management.

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

Prager J, Radev D and Czuba K (2001) Answering what-is questions by virtual annotation. In: Proceedings of Human Language Technologies Conference, pp. 26–30.

Radev D, Fan W, Qi H, Wu H and Grewal A (2002) Probabilistic question answering on the web. In: 2002 WWW conference.

Ratnaparkhi A (1996) A maximum entropy model for part-of-speech tagging. In: Brill E and Church K, eds., Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 133–142.

Rosenfeld R (1996) A maximum entropy approach to adaptive statistical language modeling. Computer, Speech, and Language, 10:187–228.

Vapnick V (1998) Statistical Learning Theory. John Wiley & Sons.

Voorhees E and Tice D (1999) The TREC-8 question answering track evaluation. In: Proceedings of the 8th Text Retrieval Conference (TREC-8).

Voorhees EM (2000) overview of the trec-9 question answering track. In: Proceedings of the 9th Text Retrieval Conference (TREC-9).

Voorhees EM (2001) Overview of the TREC 2001 question answering track. In: Proceedings of the 10th Text Retrieval Conference (TREC 2001).

Yang Y and Liu X (1999) A re-examination of text categorization methods. In: Hearst MA, Gey F and Tong R eds. in Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval. Berkeley, US, pp. 42–49.

Zhang D and Lee WS (2003) Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 26–32.