A Survey of State-of-the-Art Methods on Question Classification

Babak Loni

Delft University of Technology, Mediamatics Department,
PO Box 5031, 2600 GA Delft, Netherlands
b.loni@student.tudelft.nl

Abstract. The task of question classification (QC) is to predict the entity type of a question which is written in natural language. This is done by classifying the question to a category from a set of predefined categories. Question classification is an important component of question answering systems and it attracted a notable amount of research since the past decade. This paper gives a through overview of the state-of-the-art approaches in question classification and provides a detailed comparison of recent works on question classification and discussed about possible extensions to QC problem.

1 Introduction

By the rapidly increasing amount of knowledge in the Web, search engines need to be more intelligent than before. In many cases the user only needs a specific piece of information instead of a list of documents. Rather than making the user to read the entire document, it is often preferred to give the user a concise and short answer. Question Answering (QA) systems are aimed to provide the exact piece of information in response to a question. An *open domain* question answering system should be able to answer a question written in natural language, similar to humans.

The study to build a system which answers natural language questions backs to early 1960s. The first question answering system, BASEBALL, (Green et al., 1961) was able to answer *domain-specific* natural language questions which was about the baseball games played in American league over one season. This system was simply a database-centered system which used to translate a natural language question to a canonical query on database.

Most of other early studies (Simmons, 1965; Woods, 1973; Lehnert, 1977) was mainly domain-specific systems or have many limitation on answering questions. Due to lack of enough back-end knowledge to provide answer to open domain questions, the research on question answering systems lay dormant for few decades until the emergence of the web. The huge amount of data on the web on one hand and the need for querying the web on the other hand, brought again the task of question answering into focus. The focus on question answering research increased specially when the Text REtrieval Conference (TREC) began a QA track in 1999 (Voorhees and Harman, 2000).

The simplest type of question answering systems are dealing with *factoid questions* (Jurafsky and Martin, 2008). The answer of this type of questions are simply one or more words which gives the precise answer of the question. For example questions like "What is a female rabbit called?" or "Who discovered electricity?" are factoid questions. Sometimes the question asks for a body of information instead of a fact. For example questions like "What is gymnophobia ?" or "Why did the world enter a global depression in 1929?" are of these type. To answer these questions typically a *summary* of one or more documents should be given to the user.

Many techniques from information retrieval, natural language processing and machine learning have been employed for question answering systems. Some early studies were mainly based on querying structured data while the others used to apply pattern matching techniques. Androutsopoulos et al. (1995) provides an overview of the early question answering systems. Recent studies on open-domain QA systems are typically based on Information Retrieval (IR) techniques. The IR-based question answering systems try to find the answer of a given question by processing a corpus of documents, usually from the web, and finding a segment of text which is likely to be the answer of that question.

Some other recent works are founded on some pre-defined ontologies. These systems are based on *semi-structured* knowledge-bases and can not directly process free form documents on the web. They often demand the web documents to be represented in structured or semi-structured formats. *Semantic web* (Berners-Lee et al., 2001) was the most successful attempt to represent the web documents in a structured way; although it never achieved its desired state (Anderson, 2010). Systems such as START (Katz et al., 2002), and True Knowledge[1] are two question answering engines working on top of *semi-structured* data and semantic-web-based technologies. These systems have their own knowledge bases which are mainly created by semi-automated data annotation.

What is referred as a true *automated* question answering system, is an IR-based system which can understand natural language question, process free form text and extract the true answer from text documents. A QA system which finds the answers directly from documents is called *shallow* system. If the system is capable to do inference on the facts, it is referred as *deep* QA system. Majority of current research on question answering try to come up with ideas to build such an intelligent systems, either shallow systems or deep systems.

Typically an automated QA system has tree stages (Jurafsky and Martin, 2008): question processing, passage retrieval and answer processing. Figure 1 illustrates the common architecture of a factoid QA system. Below the task of each component is briefly described:

- **Question Processing:** the task of question processing is to analyze the question and create a proper IR query as well as detecting the *entity type* of the answer, a category name which specifies the type of answer. The first task is called *query reformation* and the second is called *question classifica-*

[1] www.trueknowledge.com

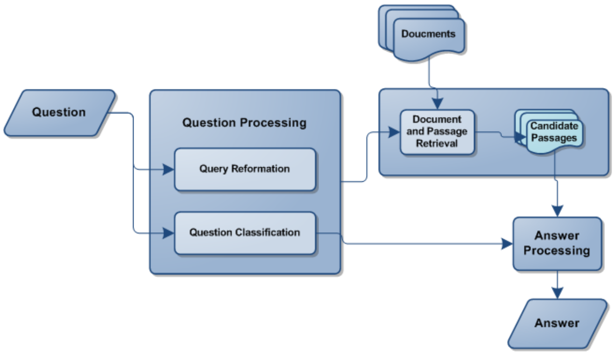


Fig. 1. The common architecture of a factoid question answering system

tion.

- **Passage Retrieval:** the task of passage retrieval is to query over the IR engine, process the returned documents and return candidate passages that are likely to contain the answer. Question classification comes handy here: it can determine the search strategy to retrieve candidate passages. Depending on the question class, the search query can be transformed into a form which is most suited for finding the answer.

- **Answer Processing:** the final task of a QA system is to process the candidate passages and extract a segment of word(s) that is likely to be the answer of the question. Question classification again comes handy here. The candidate answers are ranked according to their likelihood of being in the same class as question class and the top ranked answer(s) will be considered as the final answer(s) of the question.

In this paper we have focused on question classification, an important component of question answering systems. The task of question classification is to predict the entity type or category of the answer. This can be done by different approaches. In the next section we review the approaches on question classification. However, this paper mainly discusses the machine learning methods since most of the successful and state-of-the-art question classifiers are based on machine learning approaches.

This paper is organized as follows: in the next section we give an overview of question classification and introduce some datasets which have been used for evaluating question classifier systems. In section 3 we review successful supervised learning approaches in question classification by introducing classifiers that have been employed in learning-based QC systems. Section 4 provides a detailed review on the features which have been used for question classification together with techniques on extracting those features. We also compare successful learning-based approaches based on classifiers and features in this section.

Question classification has also been studied by using semi-supervised learning approaches. In section 5 we reviewed some successful semi-supervised studies in QC problem. We provided a review on misclassification causes as well as detailed analysis on QC performance in section 6. We finally draw conclusions and discuss about possible extensions on question classification studies, in section 7.

2 Question Classification

The task of a question classifier is to assign one or more class labels, depending on classification strategy, to a given question written in natural language. For example for the question "What London street is the home of British journalism?", the task of question classification is to assign label "Location" to this question, since the answer to this question is a named entity of type "Location". Since we predict the *type* of the answer, question classification is also referred as *answer type prediction*. The set of predefined categories which are considered as question classes usually called *question taxonomy* or *answer type taxonomy*. In this section we discuss about motivations and some basic concepts in question classification.

2.1 Why Question Classification?

Question classification has a key role in automated QA systems. Although different types of QA systems have different architectures, most of them follow a framework in which question classification plays an important role (Voorhees, 2001). Furthermore, it has been shown that the performance of question classification has significant influence on the overall performance of a QA system (Ittycheriah et al., 2001; Hovy et al., 2001; Moldovan et al., 2003).

Basically there are two main motivations for question classification: locating the answer and choosing the search strategy.

- **Locating the answer:** knowing the question class can not only reduce the search space need to find the answer, it can also find the true answer in a given set of candidate answers. For example knowing that the class of the question "who was the president of U.S. in 1934?" is of type "human", the answering system should only consider the name entities in candidate passages which is of type "human" and does not need to test all phrases within a passage to see whether it can be an answer or not.

- **Choosing search strategy:** question class can also be used to choose the search strategy when the question is reformed to a query over IR engine. For example consider the question "What is a pyrotechnic display ?". Identifying that the question class is "definition", the searching template for locating the answer can be for example "pyrotechnic display is a ..." or "pyrotechnic displays are ...", which are much better than simply searching by question words.

Even in non-IR-based QA systems, question classification have an important role. Popescu et al. (2003) for example, developed a QA system over a structured database which uses question class to generate proper SQL query over the database.

2.2 Question Classification Approaches

There are basically two different approaches for question classification: rule-based and learning based. There is also some hybrid approaches which combine rule-based and learning based approaches (Huang et al., 2008; Ray et al., 2010; Silva et al., 2011).

Rule based approaches try to match the question with some manually hand-crafted rules (Hull, 1999; Prager et al., 1999). These approaches however, suffer from the need to define too many rules (Li and Roth, 2004). Furthermore, while rule-based approaches may perform well on a particular dataset, they may have quite a poor performance on a new dataset and consequently it is difficult to scale them. Li and Roth (2004) provided an example which shows the difficulty of rule-based approaches. All the following samples are same question which has been reformulated in different syntactical forms:

- What tourist attractions are there in Reims?
- What are the names of the tourist attractions in Reims?
- What do most tourist visit in Reims?
- What attracts tourists to Reims?
- What is worth seeing in Reims?

All the above questions refer to same class while they have different syntactical forms and therefore they need different matching rules. So it is difficult to make a manual classifier with a limited amount of rules.

Learning-based approaches on the other hand, perform the classification by extracting some features from questions, train a classifier and predicting the class label using the trained classifier. Many successful learning-based classification approaches have been proposed. Later in section 3 we will discuss about learning-based approaches in more details.

There are also some studies that uses both rule-based and learning based approaches together. The study of Silva et al. (2011), which is one of the most successful works on question classification, first match the question with some pre-defined rules and then use the matched rules as features in the learning-based classifier. The same approach is used in the work by Huang et al. (2008).

Since learning-based and hybrid methods are the most successful approaches on question classification and most of the recent works are based on these approaches, in this paper we mainly review the learning and hybrid approaches of question classification.

2.3 Question Type Taxonomies

The set of question categories (classes) are usually referred as *question taxonomy* or *question ontology*. Different question taxonomies have been proposed in different works, but most of the recent studies are based on a two layer taxonomy proposed by Li and Roth (2002). This taxonomy consists of 6 coarse-grained classes and 50 fine-grained classes. Table 1 lists this taxonomy.

Table 1. The coarse and fine grained question classes.

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

There are also other well-known question taxonomies use for question classification. The taxonomy proposed by Hermjakob et al. (2002) consists of 180 classes which is the broadest question taxonomy proposed until now.

Most of the recent learning-based and hybrid approaches use the taxonomy proposed by Li and Roth (2002) since the authors published a valuable set of 6000 labeled questions. This dataset consists of two separate set of 5500 and 500 questions in which the first is used as training set and the second is used as an independent test set. This dataset[2] which first published in University of Illinois Urbana-Champaign (UIUC) usually referred as the UIUC dataset and sometimes referred as the TREC dataset since it is widely use in the Text REtrieval Conference (TREC).

Metzler and Croft (2005) enhanced UIUC taxonomy with two more classes namely *list* and *yes-no-explain*. They created a separate dataset of 250 questions collected from MadSci[3] questions archive. MadSci is a scientific website which provides a framework in which users can ask a scientific question and receive an answer from an expert.

2.4 Decision Model

Many supervised learning approaches have been proposed for question classification (Li and Roth, 2002; Blunsom et al., 2006; Huang et al., 2008). These approaches mainly differ in the classifier they use and the features they extract.

[2] http://cogcomp.cs.illinois.edu/Data/QA/QC/

[3] http://www.madsci.org/

Most of the studies assumes that a question is unambiguous, i.e., it has only one class and therefore assign the question to the most likely class. Some other studies (Li and Roth, 2002, 2004) on the other hands, have more flexible strategy and can assign multiple labels to a given question.

If the set of possible classes represented by $C = \{c_1, c_2, ..., c_n\}$ then the task of a question classifier is to assign the most likely class c_i to a question q_j if the question can only belong to one class. If a question can belong to more than one class then the decision model will be different. For example in the work of Li and Roth (2002), they rank the classes according to posterior probabilities and select top k classes as class labels of a given question. The value of k will be chosen based on the following criteria:

$$k = \min(t, 5) \text{ s.t. } \sum_{i=1}^{t} p_i \geq T \tag{1}$$

such that p_i is the posterior probability of the i-th chosen label. The indexes are set in such a way that $p_1 \geq p_2 \geq ... \geq p_n$. The parameter T is a threshold parameter in $[0, 1]$ which is chosen experimentally. In their work, Li and Roth (2002), considered T as 0.95 implying that with probability of 95% the true label of the question is one of the k chosen labels.

Most of the studies however, consider only one label for a given question ($k = 1$) (Zhang and Lee, 2003; Huang et al., 2008; Silva et al., 2011).

2.5 Performance Metrics in Question Classification

Typically, the performance of a question classifier is measured by calculating the accuracy of that classifier on a particular test set. The accuracy in question classification is defined as follow:

$$accuracy = \frac{no.\ of\ Correctly\ Classified\ Samples}{Total\ no.\ of\ Tested\ Samples} \tag{2}$$

There are also two class-specific performance metrics: *precision* and *recall*, which can be used in question classification problem. The precision and recall of a classifier on a particular class c are defined as follow:

$$precision[c] = \frac{no.\ of\ Samples\ Correctly\ Classified\ as\ c}{no.\ of\ Samples\ Classified\ as\ c} \tag{3}$$

$$Recall[c] = \frac{no.\ of\ Samples\ Correctly\ Classified\ as\ c}{Total\ no.\ of\ Samples\ in\ Class\ c} \tag{4}$$

For the systems in which a question can only have one class, a question is correctly classified if the predicted label is the same as the true label. But for the systems which allow a question to be classified in more than one class (Li and Roth, 2002, 2004), a question is correctly classified, if one of the predicted labels is the same as the true label.

3 Supervised Learning Approaches in Question Classification

Most of the recent works on question classification are based on a supervised learning method. Supervised learning approaches learn a classifier from a given training set consisting of labeled questions. Supervised methods mainly differ in the classification model and the features which are extracted from questions.

The choice of classifier highly influences the final question classifier system. Different studies choose different classifiers. Support Vector Machines (SVM), Maximum Entropy Models and Sparse Network of Winnows (SNoW) are the most widely used classifiers in question classification. Some studies used language modeling for question classification. A few studies adopted other types of classifiers. In this section we categorized different studies based on the classifiers they used and briefly describe each classifier in different subsections.

3.1 Support Vector Machines

Support vector machines are non-probabilistic learning models for classifying data. They are especially successful for high dimensional data. SVM is a linear discriminant model which tries to find a hyperplane with maximum margin for separating the classes.

Suppose we are given a training set $(\mathbf{x}_i, y_i), i = 1, ..., n$, in which $\mathbf{x}_i = (x_{i1}, ..., x_{id})$ is a d-dimensional sample and $y_i \in \{1, -1\}$ is the corresponding label. The task of a support vector classifier is to find a linear discriminant function $g(x) = w^T\mathbf{x} + w_0$, such that $w^T\mathbf{x}_i + w_0 \geq +1$ for $y_i = +1$ and $w^T\mathbf{x}_i + w_0 \leq -1$ for $y_i = -1$. Therefore we seek for a solution such that the following condition holds:

$$y_i(w^T\mathbf{x}_i + w_0) \geq 1 \qquad i = 1, ..., n \tag{5}$$

The optimal linear function is obtained by minimizing the following quadratic programming problem (Vapnik, 1995):

$$\min \ \frac{1}{2}w^T w - \sum_{i=1}^{n} \alpha_i (y_i(w^T\mathbf{x}_i + w_0) - 1) \tag{6}$$

which leads to the following solution:

$$w = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{7}$$

where $\{\alpha_i, i = 1, ..., n; \alpha_i \geq 0\}$ are Lagrange multipliers. To be able to linearly separate data, typically the feature space should be mapped to a higher dimensional space. The mapping is done with a so-called *kernel function.*

The kernel is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which takes two samples from input space and map it to a real number indicating their similarity. For all $\mathbf{x}_i$, $\mathbf{x}_j \in \mathcal{X}$, the kernel function satisfies:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \tag{8}$$

where ϕ is an explicit mapping from input space $\mathcal{X}$ to a *dot product* feature space $\mathcal{H}$ (Hofmann et al., 2008).

To apply kernel functions on SVM classifier, typically the dual form of the equation (6) is solved:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j \tag{9}$$

where $\mathbf{x}_i.\mathbf{x}_j$ is the inner product of two samples which is an implicit kernel in the equation measuring similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. This inner product can be replaced by another kernel function leading equation (9) to be in the following form:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{10}$$

There are four types of basic kernels functions: linear, polynomial, radial basis function and sigmoid. Other types of custom kernel functions can also be applied for question classification.

In QC problem, as you will see in section 4, questions are typically represented in a very high dimensional space. SVMs usually have good performance for high dimensional data. Since SVMs are only applied in two-class classification problems, typically a so-called *one-against-all* strategy is chosen when number of classes is more than two (Webb, 2002).

In QC problem a question $\mathbf{x}_i$ can be represented by:

$$\mathbf{x}_i = (w_{i1}, ..., w_{iN}) \tag{11}$$

where w_{ik} indicates the frequency of term k in question $\mathbf{x}_i$ whereas N is the total number of terms.

The linear kernel –which is implicitly used when *bag-of-word* (BOW) features are used– for two question $\mathbf{x}_i$ and $\mathbf{x}_j$ can be defined as follow:

$$K_{\text{BOW}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{N} w_{il} w_{jl} \tag{12}$$

which is simply the inner product of the two questions.

Huang et al. (2008, 2009); Silva et al. (2011) used SVM with linear kernel. Huang et al. (2008) obtained accuracy of 89.2% on fine grained and 93.4% on coarse grained classes on TREC dataset. Silva et al. (2011) obtained an accuracy of 90.8% on fine grained and 95.0% on coarse grained classes on same dataset which is the highest accuracy reported on this dataset. The differences is because of having different feature spaces. Metzler and Croft (2005) also used SVM in their work but with Radial Basis Function (RBF) kernel. They obtained various results based on different features. They reported an accuracy of 83.6% on fine and 90.2% on coarse classes of TREC dataset and also an accuracy of 73.2% on the small MadSci dataset using a combination of features.

For the task of question classification with SVMs, linear kernels have been shown to have better performance compare to other type of kernels. Xin et al. (2005) compared the accuracy of SVM classifier based on 4 different types of kernel functions. The result on fine grained classes of TREC dataset is listed in table 2. The classifiers are trained on the same feature space.

Table 2. The accuracy of SVM classifier on TREC dataset based on different kernels. The results are taken from Xin et al. (2005)

Kernel	Linear	Polynomial	RBF	Sigmoid
Accuracy	89.2%	85.2%	85.0%	85.2%

3.2 Advanced Kernel Methods

Some studies adopt SVMs with customized kernel function. Zhang and Lee (2003) defined a tree kernel which is constructed based on the syntactical structure of question. In their approach, a given question first is parsed to its syntactic tree and then the question will be represented based on some tree fragments which are subtrees of the original syntax tree. They define a custom kernel function which maps the feature vector to a higher dimension space. In section 4.2 we will further discuss the syntactical structure of a question.

A similar approach is used to define kernel function in the study of Pan et al. (2008). They defined a semantic tree kernel which is obtained by measuring the semantic similarities of tree fragments using semantic features. They reported an accuracy of 94.0% on coarse-grained classes while Zhang and Lee (2003) obtained an accuracy of 90.0% on the same dataset.

Kernel methods have also been applied in semi-supervised style. Tomas and Giuliano (2009) defined a semantic kernel for question classification which is obtained by using unlabeled text. They used *Latent Semantic Indexing* method (Deerwester et al., 1990) to reduce the feature space to much more effective space by defining a latent semantic kernel. In their approach they defined a proximity matrix of the terms by looking at co-occurrence of information in a large corpus.

The latent semantic kernel can be obtained using *singular value decomposition* (SVD). Suppose that $\mathbf{D}$ is the term-by-document matrix from Wikipedia documents corpus in which $\mathbf{D}_{i,j}$ represents the frequency of term w_i in document d_j. SVD decomposes $\mathbf{D}$ into tree matrices: $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices whose columns are eigenvectors of $\mathbf{D}\mathbf{D}^T$ and $\mathbf{D}^T\mathbf{D}$ respectively and $\mathbf{\Sigma}$ is a diagonal matrix containing the eigenvalues of $\mathbf{D}\mathbf{D}^T$ in the diagonal. The proximity matrix in the reduced space can be obtained as follow:

$$\mathbf{\Pi} = \mathbf{U}_k \mathbf{\Sigma}_k^{-1} \tag{13}$$

where $\mathbf{U}_k$ is a $N \times k$ matrix containing the first k column of $\mathbf{U}$ and $\mathbf{\Sigma}_k$ is the diagonal matrix of corresponding eigenvalues. The proximity matrix $\mathbf{\Pi}$ can be used to define a transformation $\pi : \mathbb{R}^N \to \mathbb{R}^k$ which maps a question $\mathbf{x}_i$ to the vector $\acute{\mathbf{x}}_i$ as follow:

$$\pi(\mathbf{x}_i) = \mathbf{x}_i(\mathbf{W\Pi}) = \acute{\mathbf{x}}_i \tag{14}$$

where $\mathbf{W}$ is a $N \times N$ diagonal matrix in which $\mathbf{W}_{i,i} = idf(w_i)$ is the *inverse document frequency (idf)* of the term w_i. The function *idf* reflects the importance of a word by measuring how frequent that word appears in the document corpus. It is assumed that the words which are recurred more often are less important and have lower *idf* value and the words which appear very few, are more important and have higher *idf* value. Tomas and Giuliano (2009) obtained the *idf* values of the words by collecting 50,000 randomly collected Wikipedia pages. According to the above equation, the latent semantic kernel can be defined as follow:

$$K_{LS}(\mathbf{x}_i, \mathbf{x}_j) = \langle \pi(\mathbf{x}_i), \pi(\mathbf{x}_j) \rangle \tag{15}$$

In their experiment, Tomas and Giuliano (2009) reduced the feature space to 400 dimensions by setting k to 400. They also defined a semantic kernel function based on a manually constructed list of related words. The semantic related kernel K_{Rel} is defined as follow:

$$K_{Rel}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \mathbf{P}\mathbf{P}^T \mathbf{x}_j^T = \acute{\mathbf{x}}_i \acute{\mathbf{x}}_j^T \tag{16}$$

where $\mathbf{P}$ is proximity matrix which reflects the similarity between the words in the list. Tomas and Giuliano (2009) do their experiment on TREC dataset by applying different kernels on the input feature space. Table 3 lists the accuracy of their experiment on TREC dataset. The best result is obtained by combination of all three kernels.

Table 3. The accuracy of kernel methods on TREC dataset based on different kernel functions. The results are taken from Tomas and Giuliano (2009)

Kernel	**Accuracy**	
	Coarse	Fine
K_{BOW}	86.4%	80.8%
K_{LS}	70.4%	71.2%
$K_{\text{BOW}} + K_{LS}$	90.0%	83.2%
$K_{\text{BOW}} + K_{Rel}$	89.4%	84.0%
$K_{\text{BOW}} + K_{LS} + K_{Rel}$	90.8%	85.6%

3.3 Maximum Entropy Models

Maximum Entropy (ME) models which are also known as Log Linear models is another successful classifier used in question classification. In contrast to SVMs, maximum-entropy model is an statistical approach which can calculate the probability of belonging to each class for a given sample. Additionally, ME models can be used for multiple class assignment strategy (see equation 1) while SVMs can only be used for single class assignment. Furthermore the uncertainty of the assigned label can be used later to rank the final answer.

ME models are very useful when there are many overlapping features, i.e., when the features are highly correlated. In the case of question classification as you will see in the next section, it often happens that features are very dependent.

In ME model the probability that sample $\mathbf{x}_i$ belongs to class y_j is calculated as following (Berger et al., 1996):

$$p(y_j|\mathbf{x}_i, \lambda) = \frac{1}{Z(\mathbf{x}_i|\lambda)} \exp \sum_{k=1}^{n} \lambda_k f_k(\mathbf{x}_i, y_j) \tag{17}$$

where f_k is feature indicator function which is usually binary-valued function defined for each feature; λ_k is weight parameter which specifies the importance of $f_k(\mathbf{x}_i, y_j)$ in prediction and $Z(\mathbf{x}_i|\lambda)$ is a normalization function which is determined by the requirement $\sum_j p(y_j|\mathbf{x}_i, \lambda) = 1$ for all $\mathbf{x}_i$:

$$Z(\mathbf{x}_i|\lambda) = \sum_j \exp \sum_{k=1}^{n} \lambda_k f_k(\mathbf{x}_i, y_j) \tag{18}$$

Typically, in question classification f_k is a binary function of questions and labels and defined by conjunction of class label and predicate features (Blunsom et al., 2006). The following equation is a sample of feature indicator function in question classification:

$$f_k(\mathbf{x}, y) = \begin{cases} 1 \text{ if word who in } \mathbf{x} \text{ \& } y\text{=HUM:individual} \\ 0 \text{ otherwise} \end{cases} \tag{19}$$

To learn the parameters of the model (λ), ME tries to maximize the log-likelihood of the training samples:

$$\mathcal{LL} = \sum_i \log \frac{\exp \sum_{k=1}^{N} \lambda_k f(\mathbf{x}_i, y_i)}{\sum_j \exp \sum_{k=1}^{N} \lambda_k f(\mathbf{x}_i, y_j)} \tag{20}$$

where N is number of features, $\mathbf{x}_i$ is the i^{th} training sample, y_i is its label respectively. To avoid overfitting in ME model, usually a prior distribution of the model parameters is also added to the above equation. Blunsom et al. (2006) defined a Gaussian prior in their model:

$$p(\lambda_k) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\lambda_k^2}{2\sigma^2}) \tag{21}$$

By considering the Gaussian prior, the log-likelihood objective function will be as follows:

$$\mathcal{LL} = \sum_{i=1}^{n} \log \frac{\exp \sum_{k=1}^{N} \lambda_k f_k(\mathbf{x}_i, y_i)}{\sum_j \exp \sum_{k=1}^{N} \lambda_k f_k(\mathbf{x}_i, y_j)} + \sum_{k=1}^{n} \log p(\lambda_k) \tag{22}$$

The optimal parameters of the model (λ) will be obtained by maximizing the above equation.

Several studies adopted ME model in their work. Kocik (2004) did his experiment on TREC dataset and obtained accuracy of 85.4% on fine and 89.8% on coarse-grained classes. By extracting better features, Blunsom et al. (2006) reached an accuracy of 86.6% on fine-grained and 92.0% on coarse-grained classes on same dataset. In more recent work, Huang et al. (2008) yet obtained better results due to better feature extraction techniques. They reached an accuracy of 89.0% on fine and 93.6% on coarse-grained classes on the same dataset.

Le Nguyen et al. (2007) proposed a sub-tree mining approach for question classification. In their approach a question is parsed and the subtrees of the parsed tree is considered as features. They used ME model for classification and reported an accuracy of 83.6% on the fine-grained classes of TREC dataset. They used more compact feature space compare to other works. With same feature space their result outperforms the SVM with tree kernel (Zhang and Lee, 2003).

3.4 Sparse Network of Winnows

Sparse Network of Winnows (SNoW) is a multi-class learning architecture which is specially useful for learning in high dimensional space (Roth, 1998). It learns separate linear function for each class. The linear functions are learned by an update rule. Several update rules such as naive Bayes, Perceptron and Winnow (Littlestone, 1988) can be used to learn the linear functions.

Li and Roth (2002, 2004) used SNoW architecture to learn a question classifier. They introduced a hierarchical classifier which first assign a coarse label to a question and then uses the assigned label together with other features, as input features for the next level classifier.

Similar to ME model, SNoW can assign density values (probabilities) to each class for a given sample and therefore make it possible to assign multiple labels to a given sample (equation 1). Li and Roth (2002, 2004) used the multiple class assignment strategy in their model. They used same model in both studies but in the latter they extracted reacher semantic features. They obtained an accuracy of 89.3% on fine-grained classes of TREC dataset in the latter work. They also reported an accuracy of 95.0% on fine and 98.0% on coarse-grained classes when multiple labels can be assigned to a question according to decision model in equation 1.

3.5 Language Modeling

The basic idea of language modeling is that every piece of text can be viewed as being generated by a language. Language modeling have been widely used for document classification (Ponte and Croft, 1998; Jurafsky and Martin, 2008). The idea is that a document D is viewed as a sequence $w_1, ..., w_N$ of words and the probability of generating this sequence is calculated for each class. The class label is determined using the Bayes rule.

Same idea have been used for question classification (Li, 1999; Murdock and Croft, 2002; Merkel and Klakow, 2007). A question $\mathbf{x}$ can be viewed as a sequence $w_1, ..., w_m$ of words such that w_i is the i^{th} word in the question. In fact a question can be viewed as a *mini-document.* The probability of generating question $\mathbf{x}$ by a language model given class c, can be calculated as follow:

$$p(\mathbf{x}|c) = p(w_1|c)p(w_2|c, w_1)...p(w_n|c, w_1, ..., w_{m-1}) \tag{23}$$

such that $p(w_i|c, w_1, ..., w_{i-1})$ is the probability that the word w_i appears after the sequence of $w_1, ..., w_{i-1}$ given class c. Since learning all these probabilities needs a huge amount of data usually a *unigram* assumption is made to calculate the probabilities, i.e., the probability of appearing w_i in a question only depends on the immediate words before w_i. Applying this assumption to (23) will lead to the following simpler form:

$$p(\mathbf{x}|c) = \prod_{i=1}^{m} p(w_i|c, w_{i-1}) \tag{24}$$

The most probable label is determined by applying the Bayes rule:

$$\hat{c} = \arg\max_{c} p(\mathbf{x}|c)p(c) \tag{25}$$

where $p(c)$ is the prior probability of class c which usually is calculated as a unigram language model on the specific class c (Merkel and Klakow, 2007) or can simply be considered equal for all classes (Zhai and Lafferty, 2001).

Li (1999) used this approach for question classification. He compared the result of language modeling classification with a rule based regular expression method on the old TREC dataset and the results reveal that language modeling approach perform much better than traditional regular expression method. Merkel and Klakow (2007) proposed same approach for question classification and reported an accuracy of 80.8% on TREC dataset. The main difference of language modeling method compare to other classification approaches, is that there in no need to extract complex features from a question. To obtain better results, it would be useful if the language model is trained with larger training sets.

3.6 Other Classifiers

In addition to the mentioned classifiers, other type of classifiers have also been used for question classification. Li et al. (2008) adopted the SVM together with Conditional Random Fields (CRFs) for question classification. CRFs are a type of discriminative probabilistic model which is used for labeling sequential data. In the model proposed by Li et al. (2008), a question is considered as a sequence of semantically related words. They use CRFs to label all the words in a question and the label of the *head word* is considered as the question class (head word extraction is described in section 4). Their approach differs with other question classification approaches in the sense that a question is considered as *sequential* data. Therefore it can extract features from *transition* between states as well as other common syntactic and semantic features. They reported an accuracy of 85.6% on fine-grained classes of TREC dataset.

Zhang and Lee (2003) compared the accuracy of question classification by 5 different classifiers on same feature space. They compared SVMs with Nearest Neighbor (NN), Naive Bayes (NB), Decision Tree (DT) and SNoW among which SVM performed the best. Their results on fine-grained classes of TREC dataset is listed in table 4.

Table 4. The accuracy of 5 different classifiers on TREC dataset with bag-of-word features; taken from Zhang and Lee (2003)

Approach	**Accuracy(fine)**	**Accuracy(coarse)**
NN	68.4%	75.6%
NB	58.4%	77.4%
DT	77.0%	84.2%
SNoW	74.0%	66.8%
SVM	80.2%	85.8%

The results from table 4 reveal that SVMs perform better compare to other classifiers when same feature space are used. However, depending on the extracted features, other classifiers may perform better. For example SVMs perform better rather than ME when semantic features are used (Huang et al., 2008), but on the other hand ME shows better performance when syntactical sub-trees are used as features (Le Nguyen et al., 2007). Therefore no specific classifier can always be preferred to other classifiers for question classification. Depending on feature space and other parameters, the optimal classifier can be different.

3.7 Combining Classifiers

Question classification has also been studied by combining different classifiers. Combination of classifiers can be done by different approaches. Xin et al. (2005) trained four SVM classifier based on four different type of features and combined them with various strategies. They compared Adaboost, (Schapire, 1999), Neural Networks and Transition-Based Learning (TBL) (Brill, 1995) combination methods on the trained classifiers. Their result on TREC dataset reveals that using TBL combination method can improve classification accuracy upto 1.6% compare to a single classifier which is trained on all features.

4 Features in Question Classification

To train a classifier, there is always an important problem on how to extract features and decide on the optimal set of features. For the task of question classification different studies extracted various features with different approaches. The features in question classification task can be categorized into 3 different types: lexical, syntactical and semantic features. There are different approaches to extract features from a question. We tried to cover all types of features that have been used for question classification.

In question classification task, a question is represented similar to document representation in *vector space model*, i.e., a question is a vector which is described by the words inside it. Therefore a question $\mathbf{x}$ can be represented as:

$$\mathbf{x} = (w_1, w_2, ..., w_N) \tag{26}$$

where w_i is defined as the frequency of term i in question $\mathbf{x}$ whereas N is total number of terms. Due to sparseness of feature vector only non-zero valued features are kept in feature vector. Therefore the size of samples is quite small despite the huge size of feature space. All lexical, syntactical and semantic features can be added to feature space and expand the above feature vector.

4.1 Lexical Features

Lexical features of a question are generally extracted based on the *context words* of the question, i.e., the words which appear in a question. Simply considering the context words as features is called *bag-of-word* or *unigram* features. Unigram is an special case of the so-called *n-gram* features. To extract n-gram features, any n consecutive words in a question is considered as a feature. Consider for example the question "How many Grammys did Michael Jackson win in 1983 ?" from TREC dataset. The unigram features of this question is simply all the words in this question. This question can be represented as follow in unigram feature space:

$$\mathbf{x} = \{(How, 1), (many, 1), (Grammys, 1), (did, 1), (Michael, 1), (Jackson, 1), \\ (win, 1), (in, 1), (1983, 1), (?, 1)\} \tag{27}$$

where the pair is in the form $(feature, value)$. The above representation is actually same as equation 26 but only the features with non-zero values are kept in feature vector. The frequency of the words in question (feature values) can be views as a weight value which reflects the importance of a word in a question. Loni et al. (2011) exploited this characteristic to weight the features based on their importance. They combined different feature spaces with different weight values. In their approach, the weight value of a feature space is multiplied to the feature values (term frequencies). The weight values are obtained by a greedy approach.

Huang et al. (2008) compares the performance of two different classifiers, SVM and ME model, learned over n-gram features for $n = 1, 2, 3$. Table 5 lists the classification accuracy for the 6 coarse and 50 fine-grained classes of TREC dataset.

Table 5. The accuracy of SVM and ME classifiers on n-gram feature spaces for coarse and fine grained classes on TREC dataset, taken from Huang et al. (2008)

Feature Space	**Accuracy(coarse)**		**Accuracy(fine)**	
	SVM	ME	SVM	ME
unigram	88.0%	86.6%	80.4%	78.8%
bigram	85.6%	86.4%	73.8%	75.2%
trigram	68.0%	57.4%	39.0%	44.2%

The above comparison reveals that unigram features have better performance compare to bigram and trigram. That is mainly because of the sparseness of dataset. In fact if any two consecutive words are considered as a separate feature, then the feature space is much larger compare to unigram feature space and that demands larger training size. Therefore with same training set, unigrams perform better than bigrams or trigrams.

Huang et al. (2008, 2009) considers question *wh-words* as a separate feature. They adapted 8 types of wh-words, namely *what*, *which*, *when*, *where*, *who*, *how*, *why* and *rest*. For example the wh-word feature of the question "What is the longest river in the world?" is *what*. Considering wh-words as a separate feature can improve the performance of classification according to the experimental studies.

Yet another kind of lexical feature is *word shapes*. It refers to apparent properties of single words. Huang et al. (2008) introduced 5 categories for word shapes: *all digit*, *lower case*, *upper case*, *mixed* and *other*. Word shapes alone is not a good feature set for question classification, but when they combined with other kind of features they usually improve the accuracy of classification (Huang et al., 2008; Loni et al., 2011).

Blunsom et al. (2006) introduced question's length as a separate lexical feature. It is simply the number of words in a question. Table 6 lists the lexical features of the sample question "How many Grammys did Michael Jackson win in 1983 ?". The features are represented in same form as equation 27.

4.2 Syntactical Features

A different class of features can be extracted from the syntactical structure of a question. Different works extracted several syntactical features with different approaches. The most common syntactical features are Part of Speech (POS) tags and headwords.

Table 6. Example of lexical features

Feature Space	Features
unigram	{(How, 1) (many, 1) (Grammys, 1) (did, 1) (Michael, 1) (Jackson, 1) (win, 1) (in, 1) (1983, 1) (?, 1)}
bigram	{(How-many, 1) (many-Grammys, 1) (Grammys-did, 1) (did-Michael, 1) (Michael-Jackson, 1) (Jakson-win, 1) (win-in, 1) (in-1983, 1) (1983-?, 1) }
trigram	{(How-many-Grammys, 1), (many-Grammys-did, 1), ..., (in-1983-?, 1)}
wh-word	{(How, 1)}
word-shapes	{(lowercase, 4) (mixed, 4) (digit, 1) (other, 1)}
question-length	{(question-len, 10)}

POS Tags POS tags indicate the part-of-speech tag of each word in a question such as NN (Noun), NP (Noun Phrase), VP (Verb Phrase), JJ (adjective), and etc. The following example shows the question "How many Grammys did Michael Jackson win in 1983 ?" with its POS taggs:

How_WRB many_JJ Grammys_NNPS did_VBD Michael_NNP Jackson_NNP win_VBP in_IN 1983_CD ?_.

The POS tags of a question is obtained by a POS tagger (Even-Zohar and Roth, 2001). POS tagging can be done with different approaches. There are many successful learning-based approaches including unsupervised methods (Clark, 2000) and Hidden Markov Models (Schütze and Singer, 1994) with 96%-97% accuracies.

Some studies in question classification add all the POS tags of question in feature vector (Li and Roth, 2004; Blunsom et al., 2006). This feature space sometimes referred as *bag-of-*POS *tags*. Loni et al. (2011) introduced a feature namely *tagged unigram* which is simply the unigrams augmented with POS tags. Considering the tagged unigrams instead of normal unigrams can help the classifier to distinguish a word with different tags as two different features.

POS tag information can also be used for extracting semantic features. As you can see in the next section, POS tags can be used to disambiguate the meaning of a word to extract semantic features.

Head Words A head word is usually defined as the most informative word in a question or a word that specifies the object that question seeks (Huang et al., 2008). Identifying the headword correctly, can significantly improve the classification accuracy since it is the most informative word in the question. For example for the question "What is the oldest city in Canada ?" the headword is "city". The word "city" in this question can highly contribute the classifier to

Table 7. Sample question from TREC dataset together with their class label. The question's headword is identified by boldface.

Question	**Category**
What **county** is Modesto , California in ?	LOC:city
Who was **Galileo** ?	HUM:desc
What is an **atom** ?	DESC:def
What is the name of the chocolate **company** in San Francisco ?	HUM:gr
George Bush purchased a small interest in which baseball **team** ?	HUM:gr
What is Australia 's national **flower** ?	ENTY:plant
Why does the moon turn orange ?	DESC:reason
What is autism ?	DESC:def
What **city** had a world fair in 1900 ?	LOC:city
What is the average **weight** of a Yellow Labrador ?	NUM:weight
Who was the first **man** to fly across the Pacific Ocean ?	HUM:ind
What day and **month** did John Lennon die ?	NUM:date
What is the life **expectancy** for crickets ?	NUM:other
What **metal** has the highest melting point ?	ENTY:substance
Who developed the **vaccination** against polio ?	HUM:ind
What is epilepsy ?	DESC:def
What **year** did the Titanic sink ?	NUM:date
What is a biosphere ?	DESC:def
What **river** in the US is known as the Big Muddy ?	LOC:other
What is the **capital** of Yugoslavia ?	LOC:city

classify this question as "LOC:city". Table 7 lists 20 sample questions from TREC dataset together with their class label. The headwords are identified by boldface. This table shows the strong relation between headwords and class label. As you might see there is no suitable headword for questions of type "Definition" or "reason".

Extracting question's headword is quite a challenging problem. The headword of a question usually extracted based on the syntactical structure of the question. To extract the headword we first need to parse the question to form the *syntax tree*. The syntax(parse) tree is a tree that represents the syntactical structure of a sentence base on some grammar rules. For natural language sentences written in English language, English grammar rules are used to create syntax tree. Figure 2 is an example of syntax tree for the question "What is the oldest city in Canada?".

There are successful parsers that can parse a sentence and form the syntax tree (Klein and Manning, 2003; Petrov and Klein, 2007). These parsers are statistical-based parsers which parse an English sentence based on *Probabilistic Context-Free Grammars* (PCFG) in which every rule is annotated with the probability of that rule being used. The rule's probabilities was learned based on a supervised approach on a training set of 4,000 parsed and annotated questions known as treebank (Judge et al., 2006). These parsers typically maintain an accuracy of more than 95%. Jurafsky and Martin (2008) provided a detailed

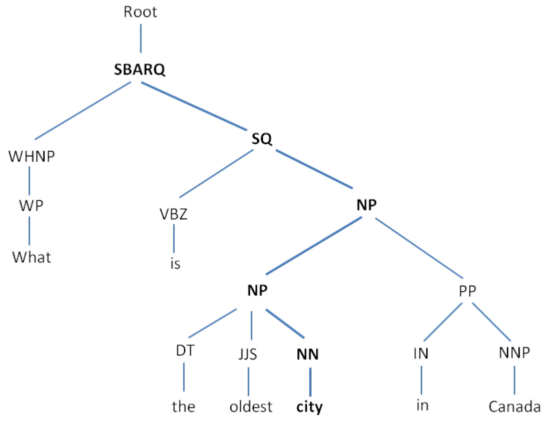


Fig. 2. The syntax tree of a sample question in which the head childs are specified by boldface

overview of parsing approaches. The list of English POS tags which is used for parsing syntax tree is listed in appendix A.

The idea of headword extraction from syntax tree first was introduced by Collins (Collins, 1999). He proposed some rules, known as Collins rules, to identify the headword of sentence. Consider a grammar rule $X \rightarrow Y_1...Y_n$ in which X and Y_i are non-terminals in a syntax tree. The head rules specifies which of the right-hand side non-terminals is the head of rule X. For example for the rule SBARQ → WHNP SQ, Collins rules specifies that the head is in the SQ non-terminal. This process continues recursively until a terminal node is reached.

To find the headword of a sentence, the parse tree is traversed top-down and in each level the subtree which contains the headword is identified with Collins head rules. The algorithm continues on the resulting subtree until it reaches a terminal node. The resulting node is the sentence's headword.

For the task of question classification, however, Collins rules are not suitable since they have preferences for verb phrases over noun phrases whereas in a question the headword should be a noun. Huang et al. (2008) and Silva et al. (2011) modified the Collins rules to properly extract a question's headword. Algorithm 1 lists the headword extraction algorithm based on Collins modified rules (Silva et al., 2011).

To follow the algorithm consider the parse tree of the question "What is the oldest city in Canada ?". The parse tree of this question is depicted in figure 2 in which the path of finding the headword is specified by boldface. The procedure Apply-Rules, finds a child of the parse tree which contains the headword, based on the modified Collins rules. Table 8 lists a subset of modified Collins rules for finding the headword of a question. The first column of the table is the non-terminal on the left side of a production rule. The second column specifies the direction of search in the right hand side of a production rule. The search can

Algorithm 1 Headword extraction algorithm

```
procedure Extract-Question-Headword (tree)
  if IsTerminal(tree) then
    return tree
  else
    head-child ← Apply-Rules(tree)
    return Extract-Question-Headword (head-child)
  end if
end procedure
```

be either by *category*, which is the default search method, or by *position*. If the direction of search is left by category then the algorithm starts from the leftmost child and check it against items in priority list (column 3 in table 8) and if it matches any, then the matched item will be returned as head. Otherwise if the algorithm reaches the end of the list and the child does not match with any of the items, it continues the same process with the next child.

On the other hand, if the search direction is left by position, then the algorithm first starts checking the items in priority list and for each item it tries to match it with every child from left to right. The first matched item is considered as head.

Now if we trace the algorithm for the sample in figure 2, it starts from top of the tree with the production rule SBARQ → WHNP SQ. The direction of search for the rule SBARQ is left by category. Therefore the algorithm starts with WHNP and check it against items in the priority list of the rule SBARQ. Because none of the items in this list match with WHNP the algorithm continues with next child. Since the next child appears in the priority list it is considered as head. With similar way the non-terminal NP will be selected in the production rule SQ → VBZ NP as the head child. The algorithm continues until it reaches the terminal node "city" and return it as the headword.

The aforementioned algorithm for extracting a question's headword can not always determine the true headword. For example for the question "Which country are Godiva chocolate from ?" the true headword is "country" while the algo-

Table 8. Modified Collins rules for determining question's headword taken from Silva et al. (2011)

Parent	**Direction**	**Priority List**
S	Left	VP, FRAG, SBAR, ADJP
SBARQ	Left	SQ, S, SINV, SBARQ, FRAG
SQ	Left	NP, VP, SQ
NP	Right by position	NP, NN, NNP, NNPS, NNS, NX
PP	Left	WHNP, NP, WHADVP, SBAR
WHNP	Left	NP
WHPP	Right	WHNP, WHADVP, NP, SBAR

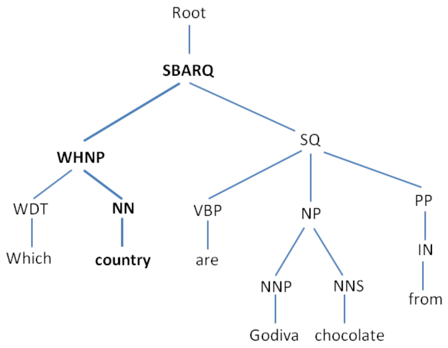


Fig. 3. The syntax tree of a sample question in which the head childs are specified by boldface. The headword in this question can not be determined correctly using the trivial rules

rithm will return “chocolate” as the headword. Figure 3 depicts the syntax tree of this question in which the head children are specified by boldface. Applying the trivial rules of algorithm 1 will choose SQ in the production rule SBARQ → WHNP SQ which leads the procedure to determine an incorrect headword.

To tackle this problem, Silva et al. (2011) introduced some *non-trivial* rules which are applied to a parse tree before applying the trivial rules. For example if SBARQ rule contains a WHXP[4] child with at least two children then WHXP is returned as head child. Considering this rule leads to correctly identifying headword in the sample of figure 3. Silva et al. (2011) reported an accuracy of 96.9% for headword extraction which is quite promising for question classification.

A question’s headword can not only be used directly as a feature, but it is also used to enhance the feature space with semantic features. Huang et al. (2008) obtained an accuracy of 81.4% on fine and 92.0% on coarse classes of TREC dataset with SVM classifier based on headword and wh-word features. This result is higher than unigram feature space in which all the words in question are considered as features. It reveals that only headword and wh-word of a question is more informative that the whole question.

Recent works (Silva et al., 2011; Loni et al., 2011) also used headwords directly and indirectly as features and their result reveals that headword is one of the most successful features in question classification.

Other Syntactic Features In addition to the mentioned syntactic features, Blunsom et al. (2006) also considered the POS tag of the headword as a separate feature. Li and Roth (2004) introduced *head chunk* as a syntactical feature. The first noun chunk and the first verb chunk after the question word are considered as head chunk. For example for the question “What is the oldest city in Canada

[4] WHXP refers to Wh-phrases: WHNP, WHPP, WHADJP, WHADVP

?" the first noun chunk is "the oldest city in Canada" since it is the first noun phrase appearing after question word.

Krishnan et al. (2005) introduced a feature namely *informer span* which is defined as short subsequent of words that are adequate clue for question classification. They extract it based on a sequential graphical model with features derived from parse tree. For example for the question "What is the tallest mountain in the world ?" the informer span is "tallest mountain". Informer span and head chunks features are added to feature vector in the same way as unigrams, i.e., all the words in head chunk or informer span are considered as a feature (table 9). Williams (2010) also considered the bigram and trigram of informer span as separate features.

Head chunks and informer span are very similar to headwords but they are usually a *sequence* of words instead of a single word. The extra words usually can introduce noisy information leading to lower accuracy rate. For example consider the question "What is a group of turkeys called ?". The headword of this question is "turkeys" while both head chunk and informer span are "group of turkeys" (Huang et al., 2008). The word "turkeys" can truly contribute to the classification of type ENTY:animal while the word group can mislead the classifier to classify this question to HUM:group. Therefore usually a single and exact headword is preferred to head chunk or informer span.

Xin et al. (2005) introduced a feature namely *words dependency* which is extracted using syntactical structure of question. Dependent words are very similar to Bigram but they are not limited to consecutive words. For example in the question "Which company created the Internet browser Mosaic ?", "Internet" and "Mosaic" are two dependent word that can not be determined by Bigram. Dependency features are treated similar to Bigram when they added to feature vector. In the mentioned example "Internet-Mosaic" is a single feature that can be added to feature vector.

Table 9 lists the syntactical features discussed in this section for the sample question "What is the oldest city in Canada ?". The features are represented same as equation 27.

4.3 Semantic Features

Semantic features are extracted based on the semantic meaning of the words in a question. Different approaches for extracting semantic features have been proposed. Most of the semantic features requires a third party data source such as WordNet (Fellbaum, 1998), or a dictionary to extract semantic information for questions. The most commonly using semantic features are *hypernyms*, *related words* and *named entities*.

Hypernyms WordNet is a lexical database of English words which provides a lexical hierarchy that associates a word with higher level semantic concepts namely *hypernyms*. For example a hypernym of the word "city" is "municipality" of which the hypernym is "urban area" and so on. As hypernyms allow one

Table 9. Example of syntactic features

Feature Space	**Features**
tagged unigram	{(What_WP, 1) (is_VBZ, 1) (the_DT, 1) (oldest_JJS, 1) (city_NN, 1) (in_IN, 1) (Canada_NNP, 1) (?_, 1)}
POS tags	{(WP, 1) (VBZ, 1) (DT, 1) (JJS, 1) (NN, 1) (IN, 1) (NNP, 1)}
headword	{(city, 1)}
headword tag	{(NN, 1)}
head chunk	{(the, 1) (oldest, 1) (city, 1)}
informer span	{(oldest, 1) (city, 1)}
words dependency	{(What-is, 1) (the-oldest, 1) (What-city, 1) (oldest-city, 1) (city-Canada, 1)}

to abstract over specific words, they can be useful features for question classification.

Extracting hypernyms however, is not straightforward. There are four challenges that should be addressed to obtain hypernym features:

1. For which word(s) in the question should we find hypernyms?
2. For the *candidate* word(s), which part-of-speech should be considered?
3. The candidate word(s) augmented with their part-of-speech may have different senses in WordNet. Which sense is the sense that is used in the given question?
4. How far should we go up through the hypernym hierarchy to obtain the optimal set of hypernyms?

To address the first challenge some studies (Huang et al., 2008, 2009; Silva et al., 2011) considered the question's headword as the candidate word to be expanded with hypernyms. Skowron and Araki (2006) considered all nouns in a question as candidate words. Loni et al. (2011) compared the classification accuracy when the feature vector is expanded with the hypernyms of all words and when it only expanded with headword's hypernyms. Their results indicate that the first experiment lead to lower accuracy since it introduce noisy information in feature vector.

For the second issue the POS tag which extracted from syntactical structure of question is considered as the target POS tag of the chosen candidate word.

To tackle the third issue, the right sense of the candidate word should be determined to be expanded with its hypernyms. For example the word "capital" with noun POS can have two different meanings. It can either interpreted as "large alphabetic character" or "a seat of government". Each sense has its own hypernyms. For example "character" is a hypernym of the first sense while "location" is a hypernym of the second sense. In the question "What is the capital of Netherlands ?" for example, the second sense should be identified.

Huang et al. (2008) adopted Lesk's Word Sense Disambiguation (WSD) algorithm to determine the true sense of word according to the sentence it appears. The Lesk's algorithm (Lesk, 1986) is a dictionary-based algorithm which works based on the assumption that words in a given context tends to share common topic. Algorithm 2 lists the adopted Lesk's WSD algorithm to determine the true sense of headword in a given question.

Algorithm 2 Adopted Lesk's WSD algorithm taken from Huang et al. (2008)

```
procedure Lesk-WSD (question, headword)
  int count ← 0
  int maxCount ← -1
  sense optimum = null
  for each sense s of headword do
    count ← 0
    for each contextWord w in question do
      int subMax ← maximum no. of common words in s definition and definition of
      any sense of w
      count ← count + subMax
    end for
    if count > maxCount then
      maxCount ← count
      optimum ← s
    end if
  end for
  return optimum
end procedure
```

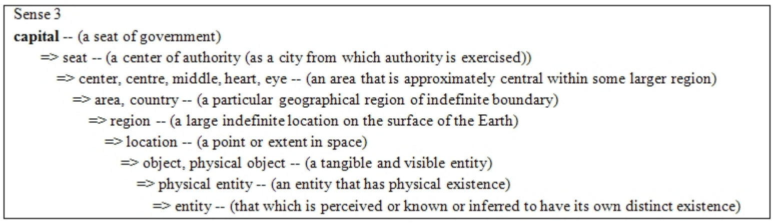
For a given headword of a question, algorithm 2 computes the maximum number of common words between the gloss (definition) of each sense and the gloss of all senses of all context words. The sense with the maximum common words is considered the true sense.

To address the fourth challenge Huang et al. (2008) considered value 6 as the maximum number of hypernyms, based on experimental results, which can be added to the feature vector, while Silva et al. (2011) considered all the hypernyms of the headword.

Consider again the question "What is the capital of Netherlands ?". The headword of this question is "capital" and the true sense of this word according to it's context is sense 3 in WordNet. Figure 4 shows the hypernym hierarchy of this sense in WordNet.

The hypernym features of this word according to representation (27) with value 6 as the maximum dept will be as follow:

{(capital, 1) (seat, 1) (center, 1) (area, 1) (region, 1) (location, 1)}



```
Sense 3
capital -- (a seat of government)
    => seat -- (a center of authority (as a city from which authority is exercised))
       => center, centre, middle, heart, eye -- (an area that is approximately central within some larger region)
          => area, country -- (a particular geographical region of indefinite boundary)
             => region -- (a large indefinite location on the surface of the Earth)
                => location -- (a point or extent in space)
                   => object, physical object -- (a tangible and visible entity)
                      => physical entity -- (an entity that has physical existence)
                         => entity -- (that which is perceived or known or inferred to have its own distinct existence)
```

Fig. 4. WordNet Hypernyms hierarchy for sense 3 of the word "capital"

The word "location" in the above features in fact can contribute the classifier to classify this question to LOC.

Related Words Li and Roth (2004) defined groups of words, each represented by a category name. If a word in the question exists in one or more groups, its corresponding categories will be added to the feature vector. For example if any of the words {birthday, birthdate, day, decade, hour, week, month, year} exists in a question, then its category name, *date*, will be added to the feature vector.

Named Entities Another semantic feature used in some studies (Li and Roth, 2004; Blunsom et al., 2006) is *named entities*. Named entities are semantic categories which can be assigned to some words in a given sentence.

Successful approaches such as Markov Models (Punyakanok and Roth, 2001) and unsupervised methods (Collins and Singer, 1999) have been employed for Named Entity Recognition (NER). Punyakanok and Roth (2001) introduced 34 semantic categories for named entity recognition and reported an accuracy of more than 90.0% on determining named entities. For example for the question "Who was the first woman killed in Vietnam War ?", their NER system identifies the following named entities: "Who was the [**number** first] woman killed in [**event** Vietnam War] ?"

In question classification the identified named entities can be added to the feature vector. Based on the representation (27) the named entity features for the aforementioned sample will be as follow: {(number, 1) (event, 1)}.

Blunsom et al. (2006) considered the named entity of the headword as a separate features due to importance of this word.

Other Semantic Features In addition to the mentioned semantic features, some studies *indirectly* used WordNet to extract semantic features. Huang et al. (2008) measured the similarity of question's headword with all question classes using WordNet hierarchy and considers the most similar category as a semantic feature.

Li and Roth (2004) uses WordNet to extract synonyms of the context words of a question and adds them to feature vector. Ray et al. (2010) uses Wikipedia to find the description of the words in a question and identifies semantic categories (named entities) of them with a rule-based algorithm

Table 10 lists the semantic features discussed in this section for the sample question "What is the oldest city in Canada ?". The features are represented same as equation (27). Note that if a feature value is larger than 1, it means that the corresponding feature is extracted from more than one word. For example in the mentioned sample there are two different words (city and Canada) both have name entity "location". Therefore the named entity features of this question will be $\{(\text{location}, 2)\}$.

Table 10. Example of semantic features

Feature Space	**Features**
headword hypernyms	{(city, 1) (municipality, 1) (urban area, 1) (geographical area, 1) (region, 1) (location, 1)}
related words	{(Rel_be, 1) (Rel_location, 2) (Rel_InOn, 1)}
named entities	{(location, 2)}
headword named entity	{(location, 1)}
indirect hypernym	{(LOC:city, 1)}

4.4 Comparison of Supervised Learning Approaches

All the methods described till now are supervised learning approaches which mainly differ in the classifier they used and the features they extract. Table 11 compares some studies on supervised learning question classification which have used TREC dataset for evaluation of their work.

From the result in table 11 it is not easy to say which classifier or which combination of features is the best choice for question classification as each method has its own advantages and disadvantages. It is however obvious that when the classifiers are trained on a richer feature space (not necessarily higher dimensional feature space), they can give a better performance. Syntactical and semantic features can usually add more information to feature space and improve classification accuracy. Since features in question classification are very dependent, usually combining all features together is not an optimal choice of features and depending on the decision model the best combination of features can be differ.

Table 11. Comparison of different supervised learning studies on question classification on TREC dataset. The abbreviation of features are:
U: Unigrams, **B**: Bigrams, **T**: Trigrams, **NG**: N-grams, **WH**: Wh-word, **WS**: Word-Shapes, **L**: Question-Length, **P**: POS-tags, **H**: Headword, **HC**: Head-Chunk, **IS**: Informer-Span, **HY**: Hypernyms, **IH**: Indirect-Hypernyms, **S**: Synonyms, **NE**: Name-Entities, **R**: Related-Words

Study	**Classifier**	**Features**	**Accuracy**	
			Coarse	Fine
Li and Roth (2002)	SNoW	U+P+HC+NE+R	91.0%	84.2%
Zhang and Lee (2003)	Tree kernel SVM	U+NG	90.0%	-
Li and Roth (2004)	SNoW	U+P+HC+NE+R+S	-	89.3%
Metzler et al. (2005)	RBF kernel SVM	U+B+H+HY	90.2%	83.6%
Krishnan et al. (2005)	Linear SVM	U+B+T+IS+HY	94.2%	88.0%
Blunsom et al. (2006)	ME	U+B+T+P+H+NE+more	92.6%	86.6%
Merkel et al. (2007)	Language Modeling	U+B	-	80.8%
Li et al. (2008)	SVM+CRF	U+L+P+H+HY+NE+S	-	85.6%
Pan et al. (2008)	Semantic tree kernel SVM	U+NE+S+IH	94.0%	-
Huang et al. (2008)	ME	U+WH+WS+H+HY+IH	93.6%	89.0%
Huang et al. (2008)	Linear SVM	U+WH+WS+H+HY+IH	93.4%	89.2%
Silva et al. (2011)	Linear SVM	U+H+HY+IH	95.0%	90.8%
Loni et al. (2011)	Linear SVM	U+B+WS+H+HY+R	93.6%	89.0%

5 Semi-Supervised Learning in Question Classification

Providing labeled questions is a costly process since it needs human effort to manually label questions while unlabeled question can be easily obtained from many web resources. Semi-supervised learning tries to exploit unlabeled information as well as labeled data. In this section we introduced semi-supervised techniques which have been used for question classification.

5.1 Co-Training

A successful semi-supervised learning algorithm which is widely used in natural language processing is *Co-training* (Blum and Mitchell, 1998). Consider we are given a training set D which consist of a labeled part $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and unlabeled part $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$. Co-training makes the strong assumption that each instance $\mathbf{x}_i$ has two views: $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}]$ such that each view consists of separate features.

The co-training algorithm trains two different classifier with labeled data for each view. The remaining unlabeled samples are classified with both classifiers and the top most-confident predictions of first view will be added to the labeled samples of second view and vise versa. The classifiers then will be re-trained and the same process continues until all unlabeled samples are used up. Algorithm 3 lists the co-training style semi-supervised learning (Zhu and Goldberg, 2009).

Algorithm 3 Co-training Algorithm

```
input labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, a learning rate $k$
  $L_1 = L_2 = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)\}$
  repeat
    train a view-1 classifier $f^{(1)}$ from $L_1$ and a view-2 classifier $f^{(2)}$ from $L_2$
    classify the remaining unlabeled data with $f^{(1)}$ and $f^{(2)}$ separately
    add top $k$ most-confident prediction $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ to $L_2$
    add top $k$ most-confident prediction $(\mathbf{x}, f^{(2)}(\mathbf{x}))$ to $L_1$
    remove the added samples from unlabeled data
  until unlabeled data is used up
```

Yu et al. (2010) applied co-training to question classification. They adopted two tree-based classifiers each of which are trained based on separate features. Their result on a Chines questions dataset reveals that under 40% rate of unlabeled data, the classification accuracy can improve up to 4 percent compare to supervised approach.

A slightly different version of co-training which have also been used in question classification is *Tri-training* (Li et al., 2008). Tri-training uses 3 classifiers instead of two. If two of the three classifiers agree to label an unlabeled instance, that instance is used for re-training other classifier.

Thanh et al. (2008) adopted tri-training for question classification. In their experiment they trained tree different classifiers: the first classifier is an SVM classifier with bag-of-word features, the second is another SVM classifier with bag-of-POS features and the third classifier is a maximum entropy classifier with both bag-of-words and bag-of-POS features. They divided TREC dataset into labeled and unlabeled parts. They compared the classification accuracy when only labeled instances are used with the situation where both labeled and unlabeled instances are used. The result shows that irrespective to the ratio of unlabeled data, the classification accuracy always increases when unlabeled samples are exploited. Table 12 lists the classification accuracy on TREC dataset when only labeled data is used and when both labeled and unlabeled data are used with tri-training algorithm.

6 Error Analysis in Question Classification

Question classification is a hard problem. As seen in previous sections, a series of complicated tasks should be done to extract proper features for a question

Table 12. Comparison of classification accuracy of supervised and semi-supervised learning based on different ratio of unlabeled data.

no. labeled	no. unlabeled	Supervised	Tri-training
1000	4452	71.0%	71.2%
2000	3452	76.4%	78.2%
3000	2452	79.0%	79.2%
4000	1452	80.8%	81.4%

and use those features in a suitable model. An incorrect decision in each of the steps can lead to a misclassification in the QC problem. However, misclassification causes is not always due to unsuitable models. Metzler and Croft (2005) explored TREC dataset and discovered 4 issues which cause misclassification in the QC problem. These issues are:

1. **Inconsistent and ambiguous labeled data**: some of the samples in TREC dataset have ambiguous label. For example the question "What does CNN stands for ?" is labeled with "ABBR:exp" while it can also be labeled by "HUM:org".

2. **Inherently difficult questions**: Some questions are even difficult for human to correctly classify them. For example consider the question "What is the name of the Lion King's son in the movie the Lion King ?". Classifying this question to type "animal" is even difficult for human.

3. **POS tagger and WordNet expansion error**: The POS taggers, parsers and WordNet expanders are not infallible and it happens that they have errors or introduce noisy information. For example consider the question "What U.S. Government agency registers trademarks ?". The POS tagger may tag this question as follow:

   "What_WP U.S._NNP Government_NN agency_NN registers_NNS trademarks_NNS ?_."

   which incorrectly tags the word "registers" as plural noun. Consequently, the headword will be misidentified to "trademarks" instead of "agency" and the incorrect headword will be expanded by WordNet which can lead to misclassification.

4. **WordNet insufficiencies**: Although expansion of the questions headword with WordNet can improve classification accuracy, it always introduces a certain amount of noise to the feature vector. Sometimes this noise can cause the question to be misclassified. For example the question "What do bats eat ?" can be correctly classified to "ENTY:food" using unigram features but when it's headword, "bats", is expanded via WordNet, it introduce noisy information which leads the question to be misclassified to "ENTY:sport".

The above misclassification causes reveal that most of the errors are due to difficulties in understanding the question. Furthermore, some types of the questions are more difficult to be correctly classified since the amount of training samples is insufficient. Loni et al. (2011) found that classifying samples of type "ENTY" and "LOC" in TREC dataset, is more difficult compare to other 4 categories of coarse grained classes. Table 13 shows the confusion matrix of the TREC dataset for coarse grained classes and table 14 lists the precision and recall of the coarse grained classes based on the results of Loni et al. (2011).

Table 13. Confusion matrix showing the classifications of the TREC dataset for the coarse categories; taken from Loni et al. (2011)

		Predicted labels					
		ABBR:*	DESC:*	ENTY:*	HUM:*	LOC:*	NUM:*
True	ABBR:*	9					
	DESC:*		134	2		1	1
	ENTY:*		10	83	1		
	HUM:*		1	1	63		
	LOC:*		1	9		71	
	NUM:*		3	2			108

Table 14. Precision and Recall of coarse grained classes of TREC dataset based on the results of Loni et al. (2011)

Class	ABBR	DESC	ENTY	HUM	LOC	NUM
Precision	100%	89.9%	85.6%	98.4%	98.6%	99.1%
Recall	100%	97.1%	88.3%	96.9%	87.6%	95.6%

Huang et al. (2008) performed a detained analysis on TREC dataset and discovered that "what" type questions are more difficult to classify compare to other type of questions. The reasons mainly back to the *ambiguity* on classifying "what" type questions which is not the case for other type of questions. For example the question "What is mad cow disease ?" can be both classified to "ENTY:disease" and "DESC:def'. Huang et al. (2008) also listed inconsistent labeling and parse errors as two other reasons of misclassifying "what" type questions. Table 15 lists the classification accuracy on TREC test set based on question types and two different classifiers. As it can be seen in this table, most of the questions are of the type "what" while they are the most difficult questions to be correctly classified.

Table 15. Classification accuracy of SVM and ME classifiers based on question types. The results are taken from Huang et al. (2008)

Question type	**#Questions**	**Coarse**		**Fine**	
		SVM	ME	SVM	ME
what	349	90.5%	91.1%	86.2%	86%
which	11	100%	100%	90.9%	100%
when	26	100%	100%	100%	100%
where	27	100%	100%	92.6%	92.6%
who	47	100%	100%	100%	100%
how	34	100%	100%	97.1%	91.2%
why	4	100%	100%	100%	100%
rest	2	100%	50.0%	0.0%	50.0%

7 Conclusion and Discussion

In this paper we presented a detailed overview on learning-based question classification approaches. Question classification is a hard problem. In fact the machine need to understand the question and classify it to the right category. This is done by a series of complicated steps. In this paper we reviewed different learning methods and feature extraction techniques for question classification. Deciding for the best model and optimal set of features is not a simple problem.

Enhancing the feature space with syntactic and semantic features can usually improve the classification accuracy. However, augmenting the feature space with more complicated features can sometimes introduce noisy information to the feature space leading to misclassification. Furthermore, due to high correlation of syntactical and semantic features, combining all possible feature spaces does not necessarily lead to higher classification accuracy.

One possible extension to the current works is to extract features in a *dynamic* way so that the questions which have enough information for classification not been expanded by more complicated features. This should be done by improving feature extraction algorithms.

Exploiting unlabeled data with semi-supervised approaches can usually improve the classification accuracy. It should be noticed that unlabeled information can sometimes introduce noisy samples to the training samples and this noise can be amplified by re-training the classifiers. Therefore semi-supervised approaches should always been used with a proper percentage of labeled and unlabeled samples on a certain amount of confident.

Feature reduction techniques such as latent semantic indexing (LSI) also have been shown to improve the performance of question classifier systems. While LSI have been largely applied in text classification, only one study used this technique for question classification. Adaptation of LSI for QC problem can also be studied in future works.

Analyzing the misclassification causes reveals that "what" type questions are typically more difficult to be classified compare to other type of questions.

More accurate feature extraction techniques are needed to deal with these type of questions. Similar to the study of Li et al. (2008), a separate classifier can be used for classifying what type questions.

Analysis of the results also reveals that some samples which are misclassified by a particular classifier can be correctly classified when another classifier is used. A *dynamic classifier selection* approach can be applied for QC problem in future works to deal with this issue.

The problem of question classification still is in the cutting edge of question answering systems. By extracting richer set of features and improving current feature extraction techniques together with more advance techniques such as semi-supervised learning, we hope that more powerful systems can be developed for question classification.

A Appendix: Part of Speech Tags

Tables 16, 17 and 18 list clause-level, phrase-level and word-level POS tags of English grammar, respectively[5]. Bies (1995) provided a detailed overview of English POS tags and their application in natural language parsing.

Table 16. The list of clauses-level POS tags

1	S	simple declarative clause
2	SBAR	Clause introduced by a (possibly empty) subordinating conjunction
3	SBARQ	Direct question introduced by a wh-word or a wh-phrase
4	SINV	Inverted declarative sentence
5	SQ	Inverted yes-no question, or main clause of a wh-question, following the wh-phrase in SBARQ

Table 17. The list of phrase-level POS tags

1	ADJP	Adjective Phrase
2	ADVP	Adverb Phrase
3	CONJP	Conjunction Phrase
4	FRAG	Fragment
5	INTJ	Interjection
6	LST	List marker
7	NAC	Not a Constituent
8	NP	Noun Phrase
9	NX	Used within certain complex NPs to mark the head of the NP
10	PP	Prepositional Phrase
11	PRN	Parenthetical
12	PRT	Particle Category for words that should be tagged RP
13	QP	Quantifier Phrase
14	RRC	Reduced Relative Clause
15	UCP	Unlike Coordinated Phrase
16	VP	Vereb Phrase
17	WHADJP	Wh-adjective Phrase
18	WHAVP	Wh-adverb Phrase
19	WHNP	Wh-noun Phrase
20	WHPP	Wh-prepositional Phrase
21	X	Unknown, uncertain, or unbracketable

[5] http://bulba.sdsu.edu/jeanette/thesis/PennTags.html

Table 18. The list of word-level POS tags

1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential ”there”
5	FW	foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	proper noun, singular
15	NNPS	proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PP	Personal pronoun
19	PP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	“to”
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rdperson singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	Wh-adverb

Bibliography

Janna Anderson. Those who understand the semantic web are split on its future, May 2010. URL `http://www.pewinternet.org/Press-Releases/2010/Semantic-Web.aspx`.

Ion Androutsopoulos, Graeme D. Ritchie, and Peter Thanisch. Natural language interfaces to databases—an introduction. *Natural Language Engineering*, 1(1): 29–81, 1995.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, 1996.

Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web (Berners-Lee et. al 2001). May 2001.

A. Bies. Bracketing Guidelines for Treebank II Style Penn Treebank Project, 1995.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM. ISBN 1-58113-057-0.

Phil Blunsom, Krystle Kocik, and James R. Curran. Question classification with log-linear models. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 615–616, New York, NY, USA, 2006. ACM.

Eric Brill. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Comput. Linguist.*, 21: 543–565, December 1995.

Alexander Clark. Inducing syntactic categories by context distribution clustering. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7*, ConLL '00, pages 91–94, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

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

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

Yair Even-Zohar and Dan Roth. A sequential model for multi-class classification. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 10–19, 2001.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press, 1998.

B.F. Green, A.K. Wolf, C. Chomsky, and K. Laughery. Baseball: An automatic question answerer. In *Proceedings Western Computing Conference*, volume 19, pages 219–224, 1961.

Kadri Hacioglu and Wayne Ward. Question classification with support vector machines and error correcting codes. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers - Volume 2*, NAACL-Short '03, pages 28–30, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

Ulf Hermjakob, Eduard Hovy, and Chin yew Lin. Automated question answering in webclopedia - a demonstration. In *In Proceedings of ACL-02*, 2002.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. July 2008.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing, 2001.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. Question classification using head words and their hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (EMNLP '08), pages 927–936, 2008.

Zhiheng Huang, Marcus Thint, and Asli Celikyilmaz. Investigation of question classifier in question answering. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, (EMNLP '09), pages 543–550, 2009.

David A. Hull. Xerox TREC-8 question answering track report. In *In Voorhees and Harman*, 1999.

A. Ittycheriah, M. Franz, W. J. Zhu, A. Ratnaparkhi, and R. J. Mammone. IBM's statistical question answering system. In *Proceedings of the 9th Text Retrieval Conference, NIST*, 2001.

John Judge, Aoife Cahill, and Josef van Genabith. Questionbank: creating a corpus of parse-annotated questions. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 497–504, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence).* Prentice Hall, 2 edition, 2008.

Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *In proceeding of the 7th international workshop on applications of natural language to information systems (NLDB)*, 2002.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *In Proceeding og the 41st annual meeting of the association for Computational Linguistic*, pages 423–430, 2003.

Krystle Kocik. Question classification using maximum entropy models. Technical report, 2004.

Vijay Krishnan, Sujatha Das, and Soumen Chakrabarti. Enhanced answer type inference from questions using sequential models. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 315–322, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Minh Le Nguyen, Thanh Tri Nguyen, and Akira Shimazu. Subtree mining for question classification problem. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1695–1700, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Wendy G. Lehnert. A conceptual theory of question answering. In *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 1*, pages 158–164, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.

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

Fangtao Li, Xian Zhang, Jinhui Yuan, and Xiaoyan Zhu. Classifying what-type questions by head noun tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 481–488, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

Wei Li. Question classification using language modeling, 1999.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, COLING '02, pages 1–7. Association for Computational Linguistics, 2002.

Xin Li and Dan Roth. Learning question classifiers: The role of semantic information. In *In Proc. International Conference on Computational Linguistics (COLING*, pages 556–562, 2004.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2:285–318, April 1988.

Babak Loni, Gijs van Tulder, Pascal Wiggers, Marco Loog, and David Tax. Question classification with weighted combination of lexical, syntactical and semantic features. In *Proceedings of the 15th international conference of Text, Dialog and Speech*, 2011.

Andreas Merkel and Dietrich Klakow. Improved methods of language model based question classification. In *In Proceedings of Interspeech Conference*, 2007.

Donald Metzler and W. Bruce Croft. Analysis of statistical question classification for fact-based questions. *Inf. Retr.*, 8:481–504, May 2005.

Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.*, 21:133–154, April 2003.

Vanessa Murdock and W. Bruce Croft. Task orientation in question answering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 355–356, New York, NY, USA, 2002. ACM.

Yan Pan, Yong Tang, Luxin Lin, and Yemin Luo. Question classification with semantic tree kernel. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 837–838, New York, NY, USA, 2008. ACM.

Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*, pages 404–411, 2007.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. pages 275–281, 1998.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI '03, pages 149–157, New York, NY, USA, 2003. ACM.

John Prager, Dragomir Radev, Eric Brown, and Anni Coden. The use of predictive annotation for question answering in trec8. In *In NIST Special Publication 500-246:The Eighth Text REtrieval Conference (TREC 8*, pages 399–411. NIST, 1999.

Vasin Punyakanok and Dan Roth. The use of classifiers in sequential inference. *Computing Research Repository*, 2001.

Santosh Kumar Ray, Shailendra Singh, and B. P. Joshi. A semantic approach for question classification using wordnet and wikipedia. *Pattern Recogn. Lett.*, 31:1935–1943, October 2010.

Dan Roth. Learning to resolve natural language ambiguities: a unified approach. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 806–813, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

Robert E. Schapire. Theoretical views of boosting and applications. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, ALT '99, pages 13–25, London, UK, 1999. Springer-Verlag. ISBN 3-540-66748-2.

Hinrich Schütze and Yoram Singer. Part-of-speech tagging using a variable memory markov model. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 181–187, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

João Silva, Luísa Coheur, Ana Mendes, and Andreas Wichert. From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review*, 35(2):137–154, February 2011.

R. F. Simmons. Answering english questions by computer: a survey. *Commun. ACM*, 8:53–70, January 1965. ISSN 0001-0782.

Marcin Skowron and Kenji Araki. Effectiveness of combined features for machine learning based question classification. *Information and Media Technologies*, 1 (1):461–481, 2006.

Nguyen Tri Thanh, Nguyen Le Minh, and Shimazu Akira. Using semi-supervised learning for question classification. *Information and Media Technologies*, 3(1): 112–130, 2008. ISSN 1881-0896.

David Tomas and Claudio Giuliano. A semi-supervised approach to question classification. In *The European Symposium on Artificial Neural Networks*, 2009.

Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

Ellen M. Voorhees. Overview of the trec 2001 question answering track. In *In Proceedings of the Tenth Text REtrieval Conference (TREC*, pages 42–51, 2001.

Ellen M. Voorhees and Donna Harman. Overview of the eighth text retrieval conference (trec-8). pages 1–24, 2000.

Andrew R. Webb. *Statistical Pattern Recognition, 2nd Edition.* John Wiley & Sons, October 2002.

Olalere Williams. High-performance question classification using semantic features. Standford University, 2010.

W. A. Woods. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, AFIPS '73, pages 441–450, New York, NY, USA, 1973. ACM.

Li Xin, HUANG Xuan-Jing, and WU Li-de. Question classification using multiple classifiers. In *Proceedings of the 5th Workshop on Asian Language Resources and First Symposium on Asian Language Resources Network*, 2005.

Zhengtao Yu, Lei Su, Lina Li, Quan Zhao, Cunli Mao, and Jianyi Guo. Question classification based on co-training style semi-supervised learning. *Pattern Recogn. Lett.*, 31:1975–1980, October 2010. ISSN 0167-8655.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 26–32, New York, NY, USA, 2003. ACM.

Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning.* Morgan & Claypool Publishers, 2009.