Subtree Mining for Question Classification Problem

Minh Le Nguyen
Japan Advanced Institute of Science and Technology
nguyenml@jaist.ac.jp

Thanh Tri Nguyen
Japan Advanced Institute of Science and Technology
t-thanh@jaist.ac.jp

Akira Shimazu
Japan Advanced Institute of Science and Technology
shimazu@jaist.ac.jp

Abstract

Question Classification, i.e., putting the questions into several semantic categories, is very important for question answering. This paper introduces a new application of using subtree mining for question classification problem. First, we formulate this problem as classifying a tree to a certain label among a set of labels. We then present a use of subtrees in the forest created by the training data to the tree classification problem in which maximum entropy and a boosting model are used as classifiers. Experiments on standard question classification data show that the uses of subtrees along with either maximum entropy or boosting models are promising. The results indicate that our method achieves a comparable or even better performance than kernel methods and also improves testing efficiency.

1 Introduction

What a current information retrieval system can do is just ”document retrieval”, i.e., given some keywords it only returns the relevant documents that contain the keywords.

However, what a user really wants is often a precise answer to a question. For instance, given the question ”Who was the first American in space?”, what a user really wants is the answer ”Alan Shepard”, but not to read through lots of documents that contain the words ”first”, ”American” and ”space” etc.

In order to correctly answer a question, usually one needs to understand what the question was asked. Question Classification can not only impose some constraints on the plausible answers but also suggest different processing strategies. For instance, if the system understands that the question ”Who was the first American in space?” asks for a person name, the search space of plausible answers will be significantly reduced. In fact, almost all the open-domain question answering systems include a question classification module. The accuracy of question classification is very important to the overall performance of the question answering system.

Question classification is a little different from document classification, because a question contains a small number of words, while a document can have a large number of words. Thus, common words like ”what”, ”else”, etc. are considered to be ”stop-words” and omitted as a dimension reduction step in the process of creating features in document. However, these words are very important for question classification. Also, word frequencies play an important role in document classification while those frequencies are usually equal to 1 in a question, thus, they do not significantly contribute to the performance of classification. Furthermore, the shortness of question makes the feature space sparse, and thus, makes the feature selection problem more difficult and different from document classification.

There are two main approaches for question classification. The first approach does question classification using handcrafted rules which are reported in [Voorhees, 1999][Voorhees, 2000][Voorhees, 2001]. Although it is possible to manually write some heuristic rules for the task, it usually requires tremendous amount of tedious work. One has to correctly figure out various forms of each specific type of questions in order to achieve reasonable accuracy for a manually constructed classification program.

The second approach applies machine learning to question classification problems such as in [Radev et al, 2002], the machine learning tool Ripper has been utilized for this problem. The author defined 17 question categories, trained and tested their question classification on TREC dataset. The reported accuracy of their system is 70%, which is not high enough for question classification. Li [Li and Roth, 2002] presented the use of SNoW method in which the good performance of their system depends on the feature called ”RelWords” (related word) which are constructed semi-automatically but not automatically. The author also reported several useful features for the question classification task including words, parts of speech, chunking labels, and named entity labels. In addition, they indicated that hierarchical classification methods is natural to the question classification problem. A question is classified by two classifiers, the first one classifies it into a coarse category, the second classifies it into a fine category.

Zhang and Lee[Zhang and Lee, 2003] employed tree kernel with a SVM classifier and indicated that the syntactic information as well as tree kernel is suitable for this problem. However, the authors showed the performance only for the coarse category, and there was no report on classifying questions to fine categories.

On the other hand, the connection of data mining techniques with machine learning problem recently shows that data mining can help improve the performance of classifier. Morhishita [Morhishita, 2002] showed that the use of association rule mining can be helpful for enhancing the accuracy of the boosting algorithm. Berzal et al also [F. Berzal et al , 2004] presented a family of decision list based on association rules mined from training data and reported a good result in the UCI data set [Li and Roth, 2002].

Mining all subtrees can be helpful for re-ranking in the syntactic parsing problem [Kudo et al, 2005] as well as ensemble learning in semantic parsing [Nguyen et al, 2006]. Along with the success, the natural question is whether or not subtree mining can be helpful for question classification problem? This paper investigates the connection of subtree mining to the maximum entropy and the boosting model. We formulate the question classification as a tree classification problem. We then present an efficient method for utilizing subtrees from the forests created by the training data in the maximum entropy model (subtree-mem) and a boosting model (subtree-boost) for the tree classification problem.

The remainder of this paper is organized as follows: Section 2 formulates the tree classification problem in which a maximum entropy model and a boosting model using subtree features are presented. Section 3 discusses efficient subtree mining methods. Section 4 shows experimental results and Section 5 gives some conclusions and plans for future work.

2 Classifier for Trees

Assume that each sentence in a question is parsed into a tree. A tree can be a sequence of word, a dependency tree, and a syntactic tree.

The problem of question classification is equivalent to classifying a syntactic tree into a set of given categories.

2.1 Definition

The tree classification problem is to induce a mapping $f(x) : X \rightarrow \{1, 2, ..., K\}$, from given training examples $T = \{\langle x_i, y_i \rangle\}_{i=1}^{L}$, where $x_i \in X$ is a labeled ordered tree and $y_i \in \{1, 2, ..., K\}$ is a class label associated with each training data. The important characteristic is that the input example xi is represented not as a numerical feature vector (bag-of-words) but a labeled ordered tree.

A labeled ordered tree is a tree where each node is associated with a label and is ordered among its siblings, that is, there are a first child, second child, third child, etc.

Let t and u be labeled ordered trees. We say that t matches u, or t is a subtree of u ($t \subseteq u$), if there exists a one-to-one function ψ from nodes in t to u, satisfying the conditions: (1)ψ preserves the parent-daughter relation, (2)ψ preserves the sibling relation, (3)ψ preserves the labels.

Let t and x be labeled ordered trees, and y be a class label ($y_i \in \{1, 2, ..., K\}$), a decision stump classifier for trees is given by:

$$h_{<t,y>}(x) = \begin{cases} 1\, t \subseteq x \\ 0\, \text{otherwise} \end{cases}$$

Figure 1 shows an example of a labeled ordered tree and its subtree and non-subtree.

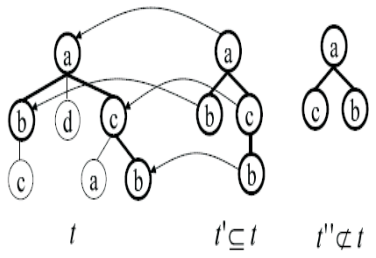

Figure 1: Labeled ordered tree and subtree relation

Decision stump functions are observed on the training data (the set of trees and their labels), then these functions are incorporated to a maximum entropy model and a boosting model.

2.2 Boosting with subtree features

The decision stumps classifiers for trees are too inaccurate to be applied to real applications, since the final decision relies on the existence of a single tree. However, accuracies can be boosted by the Boosting algorithm[Schapire, 1999] Boosting repeatedly calls a given weak learner to finally produce hypothesis f, which is a linear combination of K hypotheses produced by the prior weak learners, i,e.: $f(x) = \text{sgn}(\sum_{k=1}^{\text{K}} \alpha_k h_{<t_k, y_k>}(x))$

A weak learner is built at each iteration k with different distributions or weights $d^{(k)} = (d_i^{(k)}, ..., d_L^{(k)})$, (where $\sum_{i=1}^{N} d_i^{(k)} = 1, d_i^{(k)} \geq 0$). The weights are calculated in such a way that hard examples are focused on more than easier examples.

$$gain(< t, y >) = \sum_{i=1}^{L} y_i d_i h_{<t.y>}(x_i)$$

There exist many Boosting algorithm variants, however, the original and the best known algorithm is AdaBoost[Schapire, 1999]. For this reason we used the Adaboost algorithm with the decision stumps serving as weak functions.

2.3 Maximum Entropy Model with subtree features

Maximum entropy models [Berger et al, 1996], do not make unnecessary independence assumptions. Therefore, we are able to optimally integrate together whatever sources of knowledge we believe to be potentially useful to the transformation task within the maximum entropy model. The maximum entropy model will be able to exploit those features which are beneficial and effectively ignore those that are irrelevant. We model the probability of a class c given a vector of features x according to the ME formulation:

$$p(c|x) = \frac{\exp[\sum_{i=0}^{n} \lambda_i f_i(c, x)]}{Z_x} \quad (1)$$

Here Z_x is the normalization constant, $f_i(c, x)$ is a feature function which maps each class and vector element to a binary feature, n is the total number of features, and λ_i is a weight for a given feature function. These weights λ_i for features $f_i(c, x)$ are estimated by using an optimization technique such as the L-BFGs algorithm [Liu and Nocedal, 1989].

In the current version of maximum entropy model using subtrees, we define each feature function $f_i(c, x)$ using subtrees x and a class label c. The value of a feature function is received as +1 if (c, x) is observed in the training data. Otherwise, it is received 0. Under the current framework of maximum entropy model, all subtrees mined are incorporated to the MEM as these feature functions mentioned above.

3 Efficient subtree mining

Our goal is to use subtrees efficiently in the MEM or Boosting models. Since the number of subtrees within the corpus is large a method for generating all subtrees in the corpus seems intractable.

Fortunately, Zaki [Zaki, 2002] proposed an efficient method, rightmost-extension, to enumerate all subtrees from a given tree. First, the algorithm starts with a set of trees consisting of single nodes, and then expands a given tree of size (k .. 1) by attaching a new node to this tree to obtain trees of size k. However, it would be inefficient to expand nodes at arbitrary positions of the tree, as duplicated enumeration is inevitable. The algorithm, rightmost extension, avoids such duplicated enumerations by restricting the position of attachment. To mine subtrees we used the parameters bellow:

- minsup: the minimum frequency of a subtree in the data
- maxpt: the maximum depth of a subtree
- minpt: the minimum depth of a subtree

Table 1 shows an example of mining results.

Table 1: Subtrees mined from the corpus

Frequency	Subtree
4	(VP(VBZfeatures)(PP))
7	(VP(VP(VBNconsidered)))
2	(VP(VP(VBNcredited)))
2	(VP(VP(VBNdealt)))
5	(VP(VP(VBNinvented)))

The subtree mining using the right most extension is used in both maximum entropy model and boosting model. The following subsection will describe how these subtrees can be used in MEM and Boosting.

3.1 Subtree selection for Boosting

The subtree selection algorithm for boosting is based on the decision stump function. We will eliminate a subtree if its gain value is not appropriate for the boosting process. We borrow the definition from [Kudo et al , 2004][Kudo et al, 2005] as shown bellow:

Let $T = \{< x_1, y_1, d_1 >, ..., < x_L, y_L, d_L >\}$ be training data, where x_i is a tree and y_i is a labeled associated with x_i and d_i is a normalized weight assigned to x_i. Given T find the optimal rule $< t^0, y^0 >$ that maximizes the gain value.

The most naive and exhaustive method, in which we first enumerate all subtrees F and then calculate the gains for all subtrees, is usually impractical, since the number of subtrees is exponential to its size. The method to find the optimal rule can be modeled as a variant of the branch-and-bound algorithm, and is summarized in the following strategies:

- Define a canonical search space in which a whole set of subtrees of a set of trees can be enumerated.
- Find the optimal rule by traversing this search space.
- Prune search space by proposing a criterion with respect to the upper bound of the gain.

In order to define a canonical search space, we applied the efficient mining subtree method as described in [Zaki, 2002]. After that, we used the branch-and-bound algorithm in which we defined the upper bound for each gain function in the process of adding a new subtree. To define a bound for each subtree, we based our calculation on the following theorem [Morhishita, 2002]

For any $t' \supseteq t$ and $y \in \{1, 2, ..., K\}$, the gain of $< t', y >$ is bounded by $\mu(t)$

$$\mu(t) = \max \left\{ 2 \sum_{i|y_i=+1, t \in x_i} d_i - \sum_{i=1}^{L} y_i d_i, 2 \sum_{i|y_i=-1, t \in x_i} d_i + \sum_{i=1}^{L} y_i d_i \right\} \quad (2)$$

Note that this equation is only used for binary classification. To adapt it to the multi classification problem one can use some common strategies such as one-vs-all, one-vs-one, and error correcting code to consider the multi classification problem as the combination of several binary classifications. In this paper we focus on the use of one-vs-all for multi-class problems. Hence the subtrees can be selected by using the bound mentioned above along with binary classification using boosting.

We can efficiently prune the search space spanned by right most extension using the upper bound of gain $\mu(t)$. During the traverse of the subtree lattice built by the recursive process of rightmost extension, we always maintain the temporally suboptimal gain δ among all gains calculated previously. If $\mu(t) < delta$, the gain of any super-tree $t' \in t$ is no greater than $delta$, and therefore we can safely prune the search space spanned from the subtree t. Otherwise, we can not prune this branch. The algorithm for selecting subtrees is listed in Algorithm 1.

Note that $|t|$ means the length of a given subtree t. In addition, Algorithm 1 is presented as in an algorithm for K classes but in the current work we applied it to the binary class problem.

After training the model using subtrees, we used the one-vs-all strategy to obtain a label for a given question.

3.2 Subtree selection for MEM

Count cut-off method

The subtree selection for maximum entropy model is conducted based on a right most extension way. Each subtree is

Input: Let $T = \{< x_1, y_1, d_1 >, ..., < x_L, y_L, d_L >\}$ be training data, where x_i is a tree and y_i is a labeled associated with x_i and d_i is a normalized weight assigned to x_i.
Output: Optimal rule $< t^0, y^0 >$
Begin
Procedure project(t)
Begin
if $\mu(t) \leq \delta$ **then**
  return 1
end
$y' = \arg\max_{y \in \{1,2,...,K\}} gain(< t, y >)$ **if** $gain(< t, y' >) > \delta$ **then**
  $< t^0, y^0 >=< t, y >$
  $\delta = gain(< t, y >)$
end
for *each* $t' \in$ {*set of trees that are rightmost extension of* t } **do**
  $s =$ single node added by right most extension
  if $\mu(t) \leq \delta$ continue
  project(t');
end
End (project(t)
for $t' \in \left\{t | t \in \cup_{i=1}^{L} \{t | t \subseteq x_i\}, |t| = 1\right\}$ **do**
  project(t');
end
return $\left\langle t^0, y^0 \right\rangle$
End

Algorithm 1: Find Optimal Rule

mined using the algorithm described in [Zaki, 2002] in which a frequency, a maximum length, and minimum length of a subtree is used to select subtrees. The frequency of a subtree can suitably be used for the count cut-off feature selection method in maximum entropy models. Under the current framework of maximum entropy model we used only the simple feature selection method using count cut off method. However, the results we gained in question classification are promising.

Using boosting

In order to find effective subtree features for maximum entropy model, we applied the subtree boosting selection as presented in the previous section. We obtained all subtree features after the running of the subtrees boosting algorithm for performing on data set of one vs all strategy. In fact we obtained 50 data set for 50 class labels in the question classification data. These subtree features after applying the boosting tree algorithm in each data set will be combined to our set features for maximum entropy models. The advantage of using the boosting subtrees with maximum entropy models is that we can find an optimization combination of subtree features under the maximum entropy principle using the efficient optimization algorithm such as the L-BFGS algorithm [Liu and Nocedal, 1989]. In my opinion, the combination of subtree features using maximu entropy models might better than the linear combination as the mechanism of Adaboost.

4 Experimental results

The experiments were conducted on a pentium IV at 4.3 GHz. We tested the proposed models on the standard data similarly experiments in previous work which was collected from four sources: 5,500 English questions published by USC, about 500 manually constructed questions for a few rare classes, TREC 8 and TREC 9 questions and also 500 questions from TREC 10. The standard data is used by almost previous work on question classification [Zhang and Lee, 2003][Li and Roth, 2002]. The labels of each question is followed the two-layered question taxonomy proposed by [Li and Roth, 2002], which contains 6 coarse grained category and 50 fine grained categories, as shown bellow. Each coarse grained category contains a non-overlapping set of fine grained categories.

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

In order to obtain subtrees for our proposed methods, we initially used the Chaniak parser [E. Charniak, 2001] to parse all sentences within the training and testing data. We obtained a new training data consisting of a set of trees along with their labels. We then mined subtrees using the right most extension algorithm [Zaki, 2002]. Table 2 shows a running example of using maximum entropy models with subtrees. To train our maximum entropy model we used the L-BFGS algorithm [Liu and Nocedal, 1989] with the gaussian for smoothing.

Table 2: A running example of using maximum entropy model with subtrees

Features	Weight
ENTY (ADJP(ADVP))	0.074701
ENTY (ADJP(INof))	0.433091
ENTY (ADJP(JJcelebrated))	0.084209
HUM (ADJP(JJcommon))	0.003351
HUM (ADJP(JJdead))	0.117682
ENTY:substance (NP(ADJP(RBSmost)(JJcommon)))	0.354677
NUM:count (ADJP(PP(INfor)))	0.157310
NUM:count (NP(DTthe)(NNnumber))	0.810490

To evaluate the proposed methods in question classification, we used the evaluation method presented in [Li and Roth, 2002][Zhang and Lee, 2003] as the formula below.

$$accuracy = \frac{\#\text{of correct predictions}}{\#\text{of predictions}}$$

We conducted the following experiments to confirm our advantage of using subtree mining for classification with the maximum entropy model and boosting model. The experiments were done using significant test with 95% confident interval. Table 3 shows the accuracy of our subtree maximum entropy model (ST-Mem), subtree boosting model (ST-Boost), the combination of subtree boosting and maximum

entropy(ST-MB), and the tree kernel SVM (SVM-TK) for the coarse grained category, respectively. It shows that the boosting model achieved competitive results to that of SVM tree kernel. The ST-MB outperforms the SVM-TK and achieve the best accuracy.

Table 3: Question classification accuracy using subtrees, under the coarse grained category definition (total 6 labels)

ST-Mem	ST-Boost	ST-MB	SVM-TK
87.6	90.6	91.2	90.0

Table 4: Question classification accuracy using subtrees, under the fine grained category definition (total 50 labels).

ST-Mem	ST-Boost	ST-MB	Mem	SVM
80.5	82.6	83.6	77.6	80.2

Table 4 shows the performance of subtree-mem, subtree-boost, and subtree-MB for question classification. It also shows the result of using word features for maximum entropy models (MEM) and support vector machine models (SVM). The result of using SVM tree kernel is not reported in the original paper [Zhang and Lee, 2003]. The author did not report the performance of SVM using tree kernel on the fine grained category. Now on our paper we reported the performance of fine grained category using subtrees under the maximum entropy and the boosting model. We see that without using semantic features such as relation words we can obtain a better result in comparison with previous work. Table 4 indicates that the subtree-MB obtained the highest accuracy in comparison with other methods since it utilizes both the advantage of subtree features selection using boosting and the maximum entropy principle.

Since the subtree information and other semantic features such as the named entity types and the relation words do not overlap, mixing these features together might improve the performance of question classification task.

The results also indicate that boosting model outperformed maximum model using subtrees and the computational time of MEM is faster than that of boosting model.

The training time of maximum entropy models is approximately 12 times faster than the boosting model. We needed approximately 6 hours to finish all the training process of boosting models for 5,500 training data examples. While using the maximum entropy model we needed only approximately 0.5 hours. The computational times of combination subtree boosting features and maximum entropy models is comparable to the subtree boosting model.

The testing using MEM is also approximately 5 times faster in comparison with that of boosting model. We guess that the reason is the boosting model had to find optimal rules during the training process. In addition, the larger number of classes (50 labels) might affect to the computational time because of the strategy of one-vs-all method.

Table 5 depicted the results of boosting model and maximum model using subtrees in which the minsup and maxpt mean the frequency and the maximum depth of a subtree, respectively. This table reported that there is a relation between the subtree mined parameters and the performance of

Table 5: Question classification accuracy using subtrees, under the fine grained category definition (total 50 labels).

Parameters	Subtree-boost	Subtree-mem
maxpt=6 minsup=3	0.796	78.3
maxpt=4 minsup=2	0.826	80.50
maxpt=5 minsup=2	0.812	79.76

the boosting and maximum entropy model. So, the mathematical formulation for this relation is worth investigating.

The subtree mining approach to question classification relies on a parser to get the syntactic trees. However, most parsers, including the Charniak's parser we used in the above experiments, are not targeted to parse questions. Those parsers are usually trained on the Penn Treebank corpus, which is composed of manually labeled newswire text. Therefore it is not surprising that those parsers can not achieve a high accuracy in parsing questions, because there are only very few questions in the training corpus. In [Hermjakob, 2001], the accuracy of question parsing dramatically improves when complementing the Penn Treebank corpus [Marcus et al, 1993] with an additional 1153 labeled questions for training. We believe that a better parser is beneficial to our approach.

Summarily, all experiments show that the subtree mined from the corpus is very useful for the task of question classification. It also indicated that subtrees can be used as feature functions in maximum entropy model and weak function as in a boosting model. In addition, in the paper we give an alternative method for using subtrees in question classification. As the results showed in our experiment, we can conclude that mining subtrees is significantly contribution to the performance of question classification.

In the paper, we investigate the use of syntactic tree representation which do not overlap the named entity types and the relation words, so other tree structure representation with the combination of syntactic and semantic information be beneficial to our approach. In future, we would like to investigate the use of semantic parsing such as semantic role labelling to the task.

5 Conclusions

This paper proposes a method which allows incorporating subtrees mined from training data to the question classification problem. We formulate the question classification as the problem of classifying a tree into a prefixed categories. We then proposed the use of maximum entropy and booting model with subtrees as feature and weak functions by considering subtree mining as subtree feature selection process.

Experimental results show that our boosting model achieved a high accuracy in comparison to previous work without using semantic features. The boosting model outperformed the maximum entropy model using subtrees but its computational times is slower more than 12 times in comparison with that of the proposed maximum model. In addition, the combination of using boosting and maximum entropy models with subtree features achieved substantially bet-

ter results in comparison with other methods in term of either accuracy or computational times performance.

Future work will also be focused on extending our method to a version of using semi-supervised learning that can efficiently be learnt by using labeled and unlabeled data. We also plan to exploit the use of hierarchical classification model and WordNet semantic information as described in[Li and Roth, 2002][Li and Roth, 2006].

Furthermore, we hope to extend this work to support interactive question answering. In this task, the question answering system could be able to interact with users to lead to more variations of questions but with more contextual information.

Our subtree mining method does not depend on the task of question classification. So, we believe that our method is suitable for other tasks (i.e text classification, text summarization, etc) where its data can be represented as a tree structure.

Acknowledgments

We would like to thank to anonymous reviewers for helpful discussions and comments on the manuscript. Thanks to the Cognitive Computation Group at UIUC for opening their datasets. Thank to Mary Ann M at JAIST for correcting grammatical errors in the paper.

The work on this paper was supported by the JAIST 21 century COE program "Verifiable and Evolvable e-Society".

References

[Berger et al, 1996] Adam Berger, Stephen Della Pietra, and Vincent Della Pietra, A maximum entropy approach to natural language processing, *Computational Linguistics*, Vol. 22, No. 1, (1996).

[F. Berzal et al , 2004] F. Berzal, Juan-Carlos Cubero, Daniel Sanchez, Jose Maria Serrano. ART: A Hybrid Classification Model. *Machine learning*. Pages: 67 - 92

[E. Charniak, 2001] E. Charniak. A Maximum-Entropy Inspired Parser. *In Proc ACL 2001*.

[Carlson, 1999] Andrew Carlson, Chad Cumby and Dan Roth, The SNoW learning architecture, Technical report UIUC-DCS-R-99-2101, UIUC Computer Science Department (1999).

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik, Support vector networks, Machine Learning, Vol. 20, No. 3, pp. 273-297, (1995).

[Hacioglu and Ward, 2003] Hacioglu Kadri and Ward Wayne, Question classification with Support vector machines and error correcting codes, *proceedings of NAACL/Human Language Technology Conference*, pp. 28-30, (2003).

[Hermjakob, 2001] U. Hermjakob. Parsing and Question Classification for Question Answering. *Proceedings of the ACL Workshop on Open-Domain Question Answering*, Toulouse, France, 2001.

[Kudo et al, 2005] Taku Kudo, Jun Suzuki, Hideki Isozaki. Boosting-based parse reranking with subtree features, *Proceedings ACL 2005*.

[Kudo et al , 2004] Taku Kudo, Eisaku Maeda, Yuji Matsumoto: An Application of Boosting to Graph Classification. *Proceedings NIPS 2004*.

[Liu and Nocedal, 1989] D. Liu and J. Nocedal. On the limited memory BFGS method for large-scale optimization. *Mathematical Programming*, pp.503-528, Vol.45, 1989.

[Li and Roth, 2002] Xin Li and Dan Roth, Learning question classifiers, *Proceedings of the 19th International Conference on Computational Linguistics*, pp. 556-562, (2002).

[Li and Roth, 2006] Learning question classifiers: the role of semantic information. *Natural Language Engineering*, Volume 12, Issue 03, September 2006, pp 229-249.

[Marcus et al, 1993] M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*.

[Morhishita, 2002] Shinichi Morhishita. 2002. Computing optimal hypotheses efficiently for boosting. *In Progress in Discovery Science*, pages 471.481. Springer.

[Nguyen et al, 2006] Le-Minh Nguyen and Akira Shimazu, and Xuan-Hieu Phan. Semantic Parsing with Structured SVM Ensemble Classification Models. *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 619–626.

[Zhang and Lee, 2003] Dell Zhang and Wee Sun Lee, Question classification using Support vector machine, *proceedings of the 26th Annual International ACM SIGIR Conference*, pp. 26-32, (2003).

[Zaki, 2002] Mohammed J. Zaki. 2002. Efficiently Mining Frequent Trees in a Forest. *In proceedings 8th ACM SIGKDD 2002*.

[Schapire, 1999] Schapire, 1999. A brief introduction to boosting. *Proceedings of IJCAI 99*

[Radev et al, 2002] D. R. Radev, W. Fan, H. Qi, H. Wu and A. Grewal. Probabilistic Question Answering from the Web. *In Proceedings of the 11th World Wide Web Conference (WWW2002)*, Hawaii, 2002.

[Voorhees, 1999] E. Voorhees. The TREC-8 Question Answering Track Report. *In Proceedings of the 8th Text Retrieval Conference (TREC8)*, pp. 77-82, NIST, Gaithersburg, MD, 1999.

[Voorhees, 2000] E. Voorhees. Overview of the TREC-9 Question Answering Track. *In Proceedings of the 9th Text Retrieval Conference (TREC9)*, pp. 71-80, NIST, Gaithersburg, MD, 2000.

[Voorhees, 2001] E. Voorhees. Overview of the TREC 2001 Question Answering Track. *In Proceedings of the 10th Text Retrieval Conference (TREC10)*, pp. 157-165, NIST, Gaithersburg, MD, 2001