The Impact of Syntactic Evidence on the Effectiveness of Question Answering

Xiaoyan Li
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA
xiaoyan@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA
croft@cs.umass.edu

ABSTRACT

Syntactic information potentially plays a much more important role in question answering than it does in information retrieval. Although many people have used syntactic evidence in Question Answering, there haven't been many detailed experiments reported in the literature. The aim of the experiment described in this paper is to study the impact of a particular approach for using syntactic information on question answering effectiveness. The TREC-9 QA track data are used in the evaluation. Our results indicate that a combination of syntactic information with heuristics for ranking potential answers can perform about 8% better than the ranking heuristics on their own in terms of mean reciprocal rank and about 12% better in terms of the number of questions that correct answers are at the top rank.

Keywords: Question Answering, Syntactic Evidence.

1. INTRODUCTION

Question answering (QA) is a task different from information retrieval (IR) in that it tries to return an exact answer to short fact-based questions instead of a ranked list of documents that are likely to be relevant to users' information needs/queries. Questions submitted to QA systems are full sentences instead of 2-3 keywords typically given to web search engines. Therefore, syntactic information about how a question is phrased and how sentences in documents are structured potentially provides important clues for the matching of the question to answer candidates in the sentences.

In this paper, we present a particular approach to incorporating syntactic information in question answering. In this approach, both the question and sentences are parsed. The parser used in our system is a statistical parser (SIFT) from BBN [10]. Syntactic information is extracted from the parser output and used in the answer selection process. There are general syntactic clues that apply to all types of questions, such as matching of phrases in the question and the distance between the main verb and an answer candidate in a sentence. There are also some specific syntactic patterns that apply to different types of questions. For example, preferring an answer candidate in a possessive format in a sentence applies to "LOCATION" questions, the questions that require a location as an answer. Adjective noun phrases (NPA) which contain an answer candidate and all query words apply to "PERSON" questions, the questions that require a person name as an answer.

We have noted that other researchers have used syntactic information in their QA systems [1,3,4,5,6]. However, in addition to the differences of how to use syntactic information, as we will discuss in detail in the Related Work (Section 7), there hasn't been much detail reported on syntactic techniques in QA, especially on the impact of syntactic evidence. The most recent work [13,14] of those research groups who had used syntactic information in QA [6, 5] does not include any further report on this issue. In this paper, to study the impact of syntactic evidence on the effectiveness of question answering, a baseline QA system and a new QA system are implemented. The baseline QA system is based on the QA techniques and heuristics that are similar to that used in Li & Croft's Marsha question answering system [7]. In the new QA system, syntactic information is combined with the heuristics in the new QA system to further improve the accuracy of answer selection. Experiments are done with 162 TREC-9 questions. The questions are selected according to two criteria. First, their question-types can be determined by the question-classifier that is used in the QA system, and the expected named entities can be recognized by BBN's IdentiFinder[12], which can locate named entities such as "PERSON", "LOCATION", "TIME", "DATE", "ORGANIZATION", "MONEY" and "PERCENT" in a text file. Second, the correct answer can be found in the top 10 documents returned by INQUERY search engine. The experimental results show that the combination of heuristics and syntactic information outperform the baseline QA system which used heuristics alone.

It should be noted that in the scoring algorithm for answer selection we need to find ways to assign the weights of the heuristics and syntactic evidence that are used to calculate the score for each candidate answer. In our basic evaluation study, those weights are assigned manually according to our prior knowledge. In order to train the manual heuristic and syntactic weights used in answer selection, we have tried both maximum entropy (MaxEnt) and logistic regression. Although the learned weights in both methods have not outperformed the new QA system with the manual weights, the results provide some good indications for further research. Possible reasons for the unsatisfying results in the learning are discussed, indicating that the successful use of these techniques for QA requires appropriate features and training data.

The rest of the paper is organized as follows. Section 2 describes answer ranking in QA systems. Section 3 discusses syntactic information that can be used for QA. A particular approach of combining syntactic information with heuristics is given in Section 4. Section 5 provides the comparison experiment results, their evaluation, and some conclusions. Section 6 describes our ongoing work on applying maximum entropy methods and logistic regression techniques for learning the weights in QA scoring. Preliminary results are also reported in this section, indicating some future research directions. Related work is discussed in Section 7. Finally, conclusions and future work are given in Section 8.

2. QA WITH ANSWER RANKING

2.1 Answer Ranking

In question answering, either an answer or a ranked list of answer candidates is expected. In TREC-8 and TREC-9 QA track, a ranked list of up to five (document identifier, answer-strings) pairs for each question is required to be returned. A answer-string is limited to be at most 50 or at most 250 bytes depending on the run type. The interpretation is that answer-string is an answer to the question and doc-id is a document that provides the justification for the answer. Whether an answer or a ranked list of answer-strings is returned, answer-ranking techniques are necessary in QA systems. Typically answer candidates are sorted by their belief scores, which are calculated using heuristics or other techniques. Heuristic ranking techniques are common in QA systems. The computation of score for an answer window in the LASSO QA system by Moldovan et al. [2] considers heuristics such as the number of matching words in the passage, whether all matching words are in the same sentence, and whether the matching words in the passage have the same order as those in the question. In addition to the above heuristics, in our QA system, the size of the best matching window in a passage and the distance between an answer candidate and the center of the best matching window are considered. The best matching window of a passage here is the window that has the most query words in it and has the smallest window size.

2.2 Scoring Algorithm in Baseline QA System

The baseline QA system consists of three main components: the query processing module, the INQUERY search engine [11], and the answer extraction module. In the query processing module, each question is classified and the type of answer that this question expects is determined. A query is then generated, and is sent to the INQUERY search engine. The search engine takes the query, searches in its data collection and returns the top 10 documents that the search engine believes they are more likely to have correct answers. In the answer extraction module, answer candidates are extracted and their associated scores are calculated. An answer candidate is a named entity identified by the IdentiFinder and its type the same as the question expects. The named entity will not be considered as an answer candidate if it also appears in the question. The scoring algorithm in the baseline QA System is given in Table 1, and the heuristic score is calculated by the following equation

$$heu\_score = N + 0.5*Sm + N/W + 0.5/D \qquad (1)$$

where four heuristics are considered: the number of matching query words (*N*), whether the matching words are in the same sentence (*Sm=0/1*), the size of the best matching window (*W*), and the distance between an answer candidate and the center of the best matching window (*D*). The answer candidates then are ranked according to their scores and the answer candidate with the highest score appears at the top of the list.

Table 1. The Scoring Algorithm in the Baseline QA System

```
1. Do the following for each answer candidate in the top 10
   passages;
2. Initialize SCORE to 0;
3. Match each query word with words in each passage. Let N
   stand for the number of matching words, then
     SCORE = SCORE+N;
4. Check whether all matching words in the passage are in a single
   sentence.  If yes, then
     SCORE = SCORE +0.5;
5. Locate the best matching window in the passage and calculate
   the size of it, and the score is updated as
     SCORE = SCORE + N/size of the best matching window;
6. Locate the answer candidate in the passage and calculate the
   distance between the candidate and the center of the matching
   window in token offset. The final heuristic score is updated as
     SCORE = SCORE + 0.5/DISTANCE
```

3. SYNTACTIC INFORMATION IN QA

The heuristics used in the baseline system make no use of explicit linguistic structure. Syntactic information about how a question is phrased and how sentences in documents are structured potentially provides important clues for the matching of the question to answer candidates in the sentences.

Syntactic information can be extracted from tagging and parsing [9]. Tagging is the task of labeling each word in a sentence with its appropriate part-of-speech like noun, verb, adjective, etc. Parsing is the task of describing the structure of a sentence. The parser output is usually a tree structure with a sentence label as the root, various phrase labels as intermediate nodes, words/symbols in the sentence as leaf nodes and the parent node of a leaf node is the part-of-speech of the word in the leaf node. The parse output of a question can provide potentially more useful information than word-based approaches, where a question is simply viewed as a bag of words or limited features are considered like the order of the words in the question. Syntactic processing extracts information such as part-of-speech tags of words, phrases, and relationships between the words in the question, all of which may be useful information for QA.

In addition, from a parse tree of a sentence, noun phrases, verb phrases, and prepositional phrases etc. are easily recognized. They are usually ignored by general phrase-recognizers that mostly extract proper noun phrases and/or named entities. For example, consider question 294 from Trec9, "Who is the richest person in the world?" Figure 1 represents the parse tree of this question. From this parse tree, the phrases "the richest person" and "in the world" can be extracted. Let's consider three passages in the documents returned by INQUERY to this question, which are shown in Table 2.

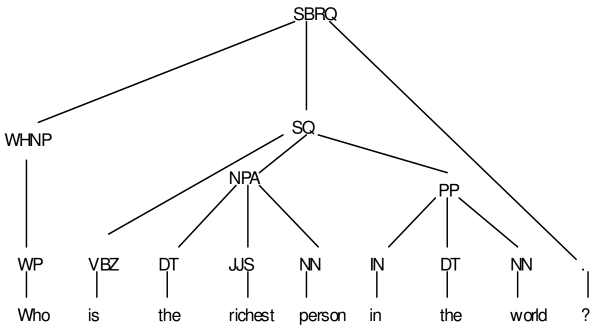

Figure 1. Parsing tree of the question "Who is the richest person in the world?" The actual output from the BBN parser we used is a string that can be easily rebuilt into the tree structure of the question.

If "the richest person" in the question is treated as single query words, then passage 1 and passage 2 may be treated as good passages and "Walton" or "Baker" may be suggested as the best answer to this question although neither of them is the correct answer. With the parse tree of the question, "the richest person" can be extracted and treated as a phrase. Passage 3 will be better than the other two passages when phrase matching is considered and the correct answer "Hassanal" may be extracted.

The main verb in the question can also be extracted given the parse tree. The relationship between "who" and the main verb in the question can be determined. It could be either active or passive. The relationship between an answer candidate and the verb in a sentence and the distance between them are also useful information in the matching of an answer context to a question. For example, for Question 631, "Who won the Nobel Prize in literature in 1988", the best passage that has the correct answer is as follows:

> *"After Naguib Mahfouz, who won the 1988 Nobel Prize in literature, Abdel-Kuddous was among the best-known novelists in the Arabic language."*

Table 2. A question and top three passages in the documents returned by INQUERY

Question	*Who is the richest person in the world?*
Passage 1	*Although tops in the U.S., Mr. Walton is the sixth-richest person in the world.*
Passage 2	*Once the richest black person in the world, Baker was destitute shortly before her death. She died in her sleep on the second night of a phenomenally successful comeback show in Paris.*
Passage 3	*As well as being the richest person in the world, Sir Hassanal lives with his relatives in the world's biggest palace _ a complex of buildings built with 38 types of marble on a 300-acre hill near the Brunei River. In case friends decide to stay over, it has 1,778 rooms and 257 toilets.*

There are two answer-candidates in this passage: "*Naguib Mahfouz*" and "*Abdel Kuddous*". "*Naguib Mahouz*" is the correct answer and "*Abdel Kuddousz*" is not. Considering the relationship between the candidate and the main verb "won" and the distance between them, *Naguib Mahouz* can be ranked as the best answer candidate, whereas "*Abdel Kuddous"* is ranked as the top of the list as the best answer candidate to this question in the baseline system which considers only the distance between the candidate and the center of the matching window.

In the new QA system, the top 10 sentences are parsed to extract syntactic information. The syntactic information is then combined with heuristics to select more likely answers. While general phrase information and verb related information applies to all types of questions, specific syntactic patterns are also considered for different types of questions. Possessive formats are detected for "LOCATION" questions. Adjective noun phrases are considered for "PERSON" questions. Whether a prepositional phrase with answer candidates modifies the main verb is considered for "LOCATION" and "DATE" questions. All syntactic information is used to adjust the belief score of answer candidates. Section 4.3 describes the details of combining syntactic information with heuristics.

4. COMBING SYNTACTIC INFO WITH HEURISTIC RANKING TECHNIQUES

In this section, we will describe in detail how syntactic information is combined with heuristics in our new QA system.

4.1 The Framework of New QA System

Figure 2 presents the relationship between the two systems. The heuristics in the baseline QA system have three main functions. First, they filter out passages that are unlikely to have correct answers. This leaves at most top 10 passages for further parsing and analyzing, thus helping speed up the run time of the new system. Second, answer candidates from the baseline system are potential "back off" answers for the new QA system. Third, the belief score of each answer candidate is a base score that will be adjusted after considering syntactic information.

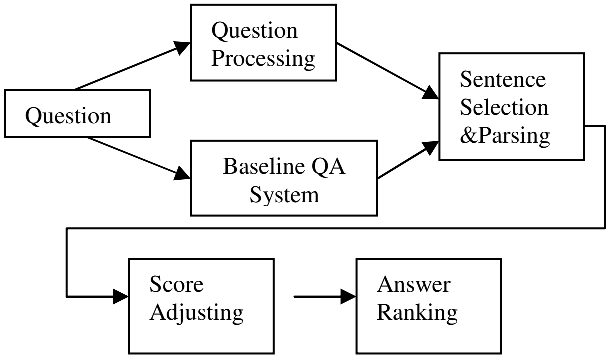

Figure 2. Framework of the New QA System

4.2 Five Steps in Our New QA system

The new QA system carries out the following five steps, which are given in figure 2:

Step 1. Question Processing. In this step, the question is parsed using BBN's SIFT. Adjective noun phrases (NPA), general noun phrases (NP) and prepositional phrases (PP) are extracted from the question. The main verb extracted is a verb in the question but not a stop word. For Who-questions asking for a person, the relationship between the word "who" and the main verb in the question is determined. It could be active or passive depending on whether the person asked is the performer of the action.

Step 2. Passage selection. The baseline QA system is used to find the top 10 passage candidates and their answer candidates. In this step, an enhancement to the original heuristics is to consider whether the candidate and all the matching words are in the same sentence, and the heuristic score is modified as the following equation

$$heu\_score^* = N + 0.5*Sm + N/W + 0.5/D + 0.5*Sc \qquad (2)$$

where Sc = 1 if the candidate and all the matching words are in the same sentence, otherwise 0.

Step 3. Sentence Selection and Parsing. From the 10 documents returned from INQUERY, one passage is selected from each document using heuristics. Each passage consists of at most 2 sentences. After considering the number of matching of unique query word and phrases, 10 sentences are selected and sent to the parser.

Step 4. Score Adjusting. In this step, syntactic information from parsing both the question and the sentences is considered and the original belief score of each answer candidate is adjusted accordingly.

Step 5. Answer Ranking. All the answer candidates are ranked by their adjusted belief scores and the top 5 answer candidates are output.

Table 3. Six syntactic factors in the new QA system

Factor 1	Match the sentence with the phrases extracted from the question. If a longer phrase is matched, then the short phrases within it will not be further considered. F1 = the size of total matched phrases/the size of the question.
Factor 2	Consider the distance between the answer candidate and the main verb in token offset. F2 = the distance between the answer candidate and the main verb.
Factor 3	For "PERSON", check the relationship between the answer candidate and the main verb in the sentence to see if it is consistent with the relationship in the question. F3 = 1 if factor 3 is satisfied, 0 otherwise.
Factor 4	For "LOCATION" questions, check the possessive formats such as, "Venezuela's Orinoco" and "Orinoco in Venezuela". F4 = 1 if factor 4 is satisfied 0 otherwise.
Factor 5	For "LOCATION" and "DATE" questions, check whether the candidate is inside a prepositional phrase and modifies the main verb. F5 = 1 if factor 5 is satisfied 0 otherwise
Factor 6	For "PERSON" questions, check whether the candidate and all query words are inside a NPA (adjective noun phrase). F4 = 6 if factor 6 is satisfied 0 otherwise.

4.3 Score Adjusting with Syntactic Information

In the step 4 described above, the original belief score of each answer candidate is adjusted. Six factors related to syntactic information are considered and the score is adjusted accordingly, which makes the final belief score for each answer candidate. Table 3 shows the six syntactic clues considered in the new QA system, and the syntactic score is calculated as

$$syn\_score=1.0*F1+0.5/F2+0.5*F3+1.0*F4+1.0*F5+1.0*F6 \qquad (3)$$

where Fi (i=1,…,6) are defined in Table 3. The weights of each factor considered in this process are currently assigned manually, based on our observations of how important the factors are. The weights are assigned 1 if we think their corresponding factors are more important. All the other weights are simply assigned 0.5. As one of the important research issues, we have used learning techniques to adjust the weights automatically based on a larger set of question and answer context pairs (Section 6). The final belief score for each candidate is then calculated using the following equation

$$\mathit{Final\_score}=\mathit{heu\_score}* + \mathit{syn\_score} \qquad (4)$$

The ranking program ranks candidates for each question by the belief score and the top 5 responses are output.

5. EXPERIMENTAL RESULT ANALYSIS

5.1 Data Collection

The baseline QA system is run on TREC9 QA track data collection that has 693 questions in total. 162 questions are finally chosen for further experiments. The questions are selected based on the following consideration. First, the question should have correct answer in the top 10 documents retrieved by INQUERY. Otherwise, it is meaningless to compare the performance of the QA systems with and without syntactic information because the correct answer doesn't appear in the top 10 documents at all. Second, the question can be classified by the question-classifier that is used in our QA system. It could be "PERSON", or "LOCATION", or "NUBMER", or "DATE", or "ORGANIZATION", or "MASS", or "MONEY", or "LENGTH", or "PERIOD" etc. Third, the named entity that the question asks for can be identified by the named-entity recognizers. Currently our data collection consists of 162 questions from TREC 9 QA questions because of the limitation of the question classifier and named entity recognizers. Nevertheless, the aim of the experiments is to study the impact of a particular approach for using syntactic information on question answering effectiveness.

5.2 Results and Evaluation

The first experiment we did is running our baseline QA system with these 162 questions. The second experiment is running the new QA system that incorporates syntactic information. Two evaluation measures are used for comparison. The first evaluation measure is the mean reciprocal answer rank from TREC-9. If the answer is found at multiple ranks, the best rank will be used. If an answer is not found in top five ranks, the score for that question is zero. With this evaluation measure, the new QA system incorporating syntactic information achieves 0.744 over 162 questions, comparing to 0.690 in the baseline QA system. The new QA system outperforms the baseline by 7.8%. The second evaluation measure is the number of questions whose correct answer can be found in the top rank. For 162 questions, there are 94 questions that the correct answer can be found in the top rank using the baseline QA system. There are 105 questions that the correct answer can be found in the top rank using the new QA system. That indicates the new QA system performs approximately 11.7% better than the baseline QA system in terms of this measure. Table 4 gives a summary of the experimental results for different types of questions, which will be further analyzed in the following subsection.

Table 4. Experimental Results

Question Type	All	Person	Location	Number[1]	Date	Organization
$N_{questions}$	162	57	56	15	25	9
M-MRR[2]	0.690	0.686	0.668	0.650	0.778	0.667
Sift-MRR[3]	0.744	0.775	0.753	0.724	0.690	0.667
Change	0.054	0089	0.085	0.074	-0.088	0
Change of %	7.8%	13.0%	12.7%	11.4%	-12.7%	0
$N_{improved}$	32	12	14	4	2	0
$N_{decreased}$	14	3	4	1	6	0

[1] **Number**: includes "NUMBER", "MASS", "MONEY", "LENGTH", and "PPERIOD" question types. [2] **M-MRR** stands for the mean reciprocal rank using Marsha heuristics alone. [3] **Sift-MRR** stands for the mean reciprocal rank incorporating syntactic information.

5.3 Analysis and Conclusions

There are three conclusions that can be drawn from the experimental results. First, the above experiment indicates that incorporating syntactic information in question answering has a positive impact on question answering effectiveness. The new QA system outperforms the baseline QA system in either of the two evaluation measures. Second, it indicates that heuristic ranking provides good back off answers for the new systems. In this sample of 162 questions, the correct answer ranks for 116 questions are unchanged. There are 46 questions whose correct answer ranks are changed, 32 of them are improved and 14 of them are decreased. Third, the impact of the syntactic evidence on the effectiveness of QA is different on different types of questions. The performance of the new QA system on "PERSON" questions, "LOCATION" questions and "NUMBER" questions are improved about 13.0%, 12.7% and 11.4% respectively. The performance on "ORGANIZATION" questions is unchanged. Surprisingly, the performance on "DATE" questions is decrease by 12.7%. The relatively higher improvements on "PERSON" and "LOCATION" questions are in our expectation because four of the six syntactic clues are applicable to these two types of questions. This indicates that the performance on other types of questions may be further improved if more specific syntactic clues are discovered and considered in QA.

To see how syntactic information works, the following is a "LOCATION" question that the correct answer is ranked to the top of all the answer candidates mainly because the syntactic clue on "LOCATION" questions is considered.

> *Question 249*: "Where is the Valley of the Kings?"
> *The sentence having the correct answer*: "The newspaper said the remains have not been disturbed since they were sent to the gardens in 1932 by Howard

Carter, who discovered the Valley of the Kings at Luxor, Egypt in 1922.”

Here “Valley of the Kings at Luxor” is detected by the system as a possessive format. The belief score of “Luxor” is then increase. That raises the rank of “Luxor” to the top of all the answer candidates in the new QA system.

The performance on 6 of 25 “DATE” questions is decreased. However, a close look at the answer context and candidates found that three of them (question 281, 471 and 625) would be actually unchanged or even improved if date information had been resolved. For question 281, the top candidate ranked by Sift score is “Sunday”. It would be a correct answer if it had been resolved to real date. For question 471, the top candidate ranked by Sift score is “April 14”. It implies a correct answer if the year information is resolved. For question 625, the top candidate is “26 April 198643”. It would be a correct answer if the “DATE” entity had been extracted correctly by IdentiFinder. If these three questions are ignored, the mean reciprocal rank over 159 questions will be increase by 0.062 and the change of percent will be 8.9%.

The following is another case in which the performance decreases:

Question 851. “When did Mount St. Helens last erupt?” Passage 1 and passage 2 are two passage candidates.

Passage 1: “"Mount St. Helens could erupt again at any time," said Don Swanson, scientist in charge at the USGS observatory in Vancouver Wash. Throughout its recorded history, Mount St. Helens has had active periods that lasted for years with relatively short spans of inactivity. Before the 1980s, the last eruptive period was from 1800 to 1857, with intermittent periods of quiet lasting months or years, according to the USGS western region office in The volcano's most recent eruptions have been quiet, dome-building affairs, in which the mountain pumps out thick lava to increase the size of the crater dome.”

Passage 2: “Mount St. Helens, historically one of the Cascade Range's most active volcanoes, had not erupted since 1857.”

“1980s” is the correct answer to this question. There are 5 query words in the question. Passage 1 has all the 5 query words and Passage 2 only has 4 of them (The query word “last” is not found in the passage). The baseline heuristics choose “1980s” in Passage 1 mainly because of the number of the matching words, while the new system chooses “1857” in the second passage. Two factors here make the belief score of “1857” in the second passage higher than “1980s” in the first passage. One is that the candidate “1857” modifies the main verb “erupt” in the second passage. The other one is that the candidate and the matching words are in the same sentence. Although “1857” is not a correct answer to this question today, it could be correct if the question was asked before 1980s. Actually, the answer to this question is time sensitive and changes when more recent information is available. This issue of time sensitivity will be considered in our future research.

6. LEARNING WEIGHTS IN QA

In the above experiments the weight associated with each factor is assigned manually according to our prior knowledge. A further research question we want to ask is: can we learn the weights automatically? Currently, we have tried maximum entropy (MaxEnt) methods and logistic regression techniques in our QA system.

6.1 Applying MaxEnt in QA

We tried MaxEnt method in QA because of the following considerations. First, MaxEnt makes no extra assumptions except all given evidence. For the distribution of the probably to be estimated, maximum entropy prefers the most uniform models that also satisfy given constraints which are represented by feature functions. Second, maximum entropy can naturally handle overlapping features and will not be hurt by strong independence assumptions. Third, maximum entropy has been used for many natural language tasks like text classification, machine translation, phrase detection, etc.

The probability model in our QA system is defined as:

$$P(x,c) = \frac{1}{Z}\prod_{j=1}^{k}\alpha_j^{f_i(x,c)} \qquad (5)$$

$$Z = \sum_{x \in X, c \in C}\prod_{j=1}^{k}\alpha_j^{f_i(x,c)} \qquad (6)$$

Here Z is a normalization factor to ensure the sum of the probabilities equal to 1. Each alpha is the weight associated with its corresponding feature. P(x, c) is the probability of seeing an answer candidate class pair. The alpha parameters can be estimated by using Generalized Iterative Scaling (GIS) on training data.

In order to apply MaxEnt method for learning the weights, all the 11 factors considered in the new QA system are transformed into binary features. As an example of the heuristic information, the value of feature No. 1 is 1 if the number of matching words/total number of query words is greater than or equal to 0.8, 0 otherwise. Another example is feature 6. it corresponds to a piece of syntactic information. The values is 1 if the number of matched phrase is greater than or equal to 1, 0 otherwise. We generated a data collection which contains 2696 answer candidates. We first transformed the 11 factors into 21 binary features. The performance is as low as 0.4 with the weight learned on the 21 features. We then combined and reduced the number of features from 21 to 11. The mean reciprocal over 162 questions is 0.568. This number is still much lower than the performance of using heuristics only. There are mainly four reasons for the poor performance of MaxEnt in our QA system: first, some useful information is lost when the heuristics and syntactic information are transformed into binary features. Second, when using Maximum entropy, the QA task is treated as a classification task. The evaluation method is more complicated. Third, our prior knowledge is that for each question, a candidate with more active features is more likely to be a correct answer. This cannot be learned from MaxEnt on our current training set. Finally, the training set is not large enough.

6.2 Applying Logistic Regression in QA

Logistic regression is another technique we have applied in our QA system. Logistic regression is a variation of ordinary regression. It is very useful when the observed outcome is restricted to two values, which usually represent the occurrence or non-occurrence of some outcome event, (correct answer or wrong answer in our task). It produces a formula that predicts the probability of the occurrence as a function of the independent variables. Since the independent variable can be continuous as well as binary, the context information (heuristics and syntactic factors) of answer candidates is not lost as it is in maximum entropy.

The probability model of logistic regression is defined as follows:

$$y = \mathrm{b}_0 + \mathrm{b}_1 * x_1 + \mathrm{b}_2 * x_2 + ... + \mathrm{b}_n * x_n \quad (7)$$

$$p = \mathrm{Exp}(y) \,/\, (\, 1 + \mathrm{Exp}(y)\,) \quad (8)$$

where p is the probability that an answer candidate is a correct answer. x_1, $x_{2,...,}$ x_n are independent variables.

Table 5. Performance using logistic regression (LR)

Performance	All	Person	Location	Number	Date	Organization
M-MRR	0.690	0.686	0.668	0.650	0.778	0.667
Sift-MRR	0.744	0.775	0.753	0.724	0.690	0.667
LR	0.717	0.678	0.763	0.606	0.753	0.759

The preliminary results with logistic regression are compared in Table 5 with QA systems having heuristics only (M-MRR) and incorporating syntactic information of manual weights (Sift-MRR). Results show that the overall performance using logistic regression is better than that of the baseline QA system, and it is very close to that of the new QA system in which the weights are assigned manually. In fact, the performance for some specific types of questions (e.g. Location and Organization) is obviously improved by learning the weights using logistic regression. Recall that currently our data collection only contains 162 questions from TREC 9 QA questions because of the limitation of the question classifier and named entity recognizers. A better performance may be achieved if a larger training data set were available. Learning different models for different types of questions with different feature combinations may also improve the performance.

7. RELATED WORK

In this section, we briefly discuss how other researchers have used syntactic information in their QA systems.

Some QA systems do not parse the sentences in documents. For example, Hull [1] used a part-of-speech tagger in his QA system. Basic keywords (e.g. who, where, how etc.) and an associated secondary argument are used to identify question type. The tagger has two functions in this QA system. First, each question is tagged for part of speech and the secondary arguments are extracted using regular expressions defined over sequences of part of speech tags. Second, the function words in the question can be identified by the tagger and then ignored in the process of sentence scoring which scores each sentence according to the number of words it has in common with the question. In Clarke et al.'s [6] QA system, only the question is parsed. The parser here has two functions. One is to generate better queries so that the passage retrieval engine can generate the best candidate passages. The other function is to generate selection rules so that the post processor can select the best 10-byt or 250-byte answers from the passages. The selection rules are patterns for given answer categories (proper, place, time etc.). These patterns generally consist of regular expressions with simple hand-coded extensions.

At the other extreme, some QA systems parse all the text in the corpus, rather than selecting a small subset of sentences that are likely to contain the answer, as is done in our system. Ferret et al.'s [3] QALC system is composed of five parallel modules and a sentences ranking module. The QALC system relies mainly on natural language processing components. Most of the components rely on a tagged version of the corpus by TreeTagger. The patterns of part of speech help assign categories to the questions in the natural language question analysis module, extract terms in the term extraction module and recognize named entities in the named entity recognition module. The parser used by Litkowski[4] is a prototype for a grammar checker. It uses a context-sensitive, augmented transition network grammar of 350 rules. Each sentence in the documents is parsed and databases are constructed by extracting relational triples from the parser output. The triples consist of discourse entities (e.g. numbers, adjective sequences, ordinals, time phrases noun constituents, etc.), semantic relations (roles as agent, theme, location, purpose, etc.), and the governing words, the words in the sentence that the discourse entity stood in relation to. Database triples are also generated for the questions. Matching between the question and sentence database records is done to find candidate sentences, which are more likely to have answers.

Harabagiu et al. [5] makes use of a statistical parser for large real-word text coverage instead of a phrasal parser. The parse trees produced by such a parser can be easily translated into a semantic form. Both the question and the paragraphs returned by the search engine are parsed and transformed into a semantic form. The WordNet semantic net is used to find lexical alternations and semantic alternations. The semantic forms of questions and answers can be unified and thus enable a matching between the conceptual relations expressed in the question and the relations derived from the answer. Our approach differs from this system in that different syntactic patterns are used for specific question types.

In summary, there are several distinctive features in our QA system. We use syntactic information from parsing the questions and sentences to select answer candidates, which are more likely to be correct answers. Heuristics are used to select up to 10 sentences for each question to be parsed. That significantly speeds up the run time. Our QA system focuses on incorporating syntactic information in answer selection. Last but not the least, we have carried out detailed experiments on the comparison of system performance with and without syntactic information.

8. CONCLUSIONS AND FUTURE WORK

Syntactic information potentially plays a much more important role in question answering than it does in information retrieval. Our experimental results indicate that a combination of syntactic information with heuristics for ranking potential answers can outperform the ranking heuristics on their own**.** The heuristics are also useful for helping filter out passages that are unlikely have correct answers, providing “back off” answers and calculating base belief scores that will be adjusted after considering syntactic information.

In the scoring algorithm for answer selection, the weights of features that are used to calculate a belief score for each candidate are assigned manually. We have tried maximum entropy methods and logistic regression techniques in our QA system to learn the weights automatically. The preliminary results using both maximum entropy and logistic regression indicate that better performance is achieved with a larger training set. Learning different models for different types of questions with different feature combinations may improve the overall performance. Logistic regression is more suitable for our problem than MaxEnt because it allows for continuous independent variables.

Another aspect of future work will focus on developing statistical models for question answering which will involve syntactic features. We have already started to develop a statistical model of question answering using the relevance-based model approach as in [8]. A dynamic aspect of question answering is also worth studying. Question 851 discussed in Section 5 is a good example for this case. For such type of questions, answers may not be decided by one document/paragraph. As new information becomes available, or as new resources are searched, answers may change or be modified.

9. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD. The grant number is N66001-99-18912. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

We also want to express our thanks to people at CIIR for their help, special thanks to David Fisher and Fangfang Feng for their valuable discussions on related research issues.

10. REFERENCES

[1] D.A. Hull, “Xerox TREC-8 Question Answering Track Report”, TREC-8, (1999).

[2] D. Moldovan et al, “LASSO: A Tool for Surfing the Answer Net,” TREC-8, pp 175-183. (1999).

[3] O. Ferret, B. Grau, G. Illouz, C. Jacquwin, and N. Masson, “QALC – the Question-Answering program of the language and Cognition group at LIMSI-CNRS”, TREC-8, (1999).

[4] K.C. Litkowski, “Question-Answering Using Semantic Relation Triples”, TREC-8, (1999).

[5] S. Harabagiu, D. Moldovan et al., “FALCON: Boosting knowledge for answer engines”, TREC-9, (2000).

[6] C.L.A. Clarke, G.G. Cormack, D.I.E. Kisman and K. Lynam, “Question Answering by Passage Selection”, TREC-9, (2000).

[7] X. Li and W.B. Croft, “Evaluating Question Answering Techniques in Chinese”, Proc. HLT 01, 96-101, (2001).

[8] V. Lavrenko and W.B. Croft, "Relevance-Based Language Models," ACM SIGIR 01, 120-127, (2001).

[9] C. D. Manning and H. Schutze, “Foundations of Statistical natural Language Processing”, The MIT Press, 1999.

[10] S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, R. Weischedel, and the Annotation Group, “Algorithms that learn to extract information--bbn: Description of the sift system as used for muc-7”. Proc. the 7th Message Understanding Conference (MUC-7). (1998).

[11] J. P. Callan, W. B. Croft, and S. M. Harding, "The INQUERY Retrieval System", Proc. the 3rd International Conference on Database and Expert Systems. (1992).

[12] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a High-Performance Learning Name-finder". Fifth Conf. on Applied Natural Language Processing (1997)

[13] Charles L. A. Clarke, Gordon V. Cormack and Thomas R. Lynam. “Exploiting Redundancy in Question Answering”, Proc. ACM SIGIR, New Orleans, September, 2001.

[14] Marius Pasca and Sanda Harabagiu “High Performance Question/Answering”, Proc. ACM SIGIR, New Orleans, September 2001.