Automatic Derivation of Surface Text Patterns for a Maximum Entropy Based Question Answering System

Deepak Ravichandran[1]
USC Information Sciences Institute
4676, Admiralty Way
Marina del Rey, CA, 90292
ravichan@isi.edu

Abraham Ittycheriah and **Salim Roukos**
IBM TJ Watson Research Center
Yorktown Heights, NY, 10598
{abei,roukos}@us.ibm.com

Abstract

In this paper we investigate the use of surface text patterns for a Maximum Entropy based Question Answering (QA) system. These text patterns are collected automatically in an unsupervised fashion using a collection of trivia question and answer pairs as seeds. These patterns are used to generate features for a statistical question answering system. We report our results on the TREC-10 question set.

1 Introduction

Several QA systems have investigated the use of text patterns for QA (Soubbotin and Soubbotin, 2001), (Soubbotin and Soubbotin, 2002), (Ravichandran and Hovy, 2002). For example, for questions like "When was Gandhi born?", typical answers are "Gandhi was born in 1869" and "Gandhi (1869-1948)". These examples suggest that the text patterns such as "<NAME> was born in <BIRTHDATE>" and "<NAME> (<BIRTHDATE> - <DEATHYEAR>)" when formulated as regular expressions, can be used to select the answer phrase to questions. Another approach to a QA system is learning correspondences between question and answer pairs. IBM's Statistical QA (Ittycheriah et al., 2001a) system uses a probabilistic model trainable from Question-Answer sentence pairs. The training is performed under a Maximum Entropy model, using bag of words, syntactic and name entity features. This QA system does not employ the use of patterns. In this paper, we explore the inclusion of surface text patterns into the framework of a statistical question answering system.

2 KM Corpus

A corpus of question-answer pairs was obtained from Knowledge Master (1999). We refer to this corpus as the KM database. Each of the pairs in KM represents a trivia question and its corresponding answer, such as the ones used in the trivia card game. The question-answer pairs in KM were filtered to retain only questions that look similar to the ones presented in the TREC task[2]. Some examples of QA pairs in KM:

1. Which country was invaded by the Libyan troops in 1983? - Chad
2. Who led the 1930 Salt March in India? - Mohandas Gandhi

3 Unsupervised Construction of Training Set for Pattern Extraction

We use an unsupervised technique that uses the QA in KM as seeds to learn patterns. This method was first described in Ravichandran and Hovy (2002). However, in this work we have enriched the pattern format by inducing specific semantic types of QTerms, and have learned many more patterns using the KM.

3.1 Algorithm for sentence construction

1. For every question, we run a Named Entity Tagger HMMNE [3] and identify chunks of words, that signify entities. Each such entity obtained from the Question is defined as a Question term (QTerm). The Answer Term (ATerm) is the Answer given by the KM corpus.
2. Each of the question-answer pairs is submitted as query to a popular Internet search engine[4]. We use the top 50 relevant documents after stripping off the HTML tags. The text is then tokenized to smoothen white space variations and chopped to individual sentences.
3. For every sentence obtained from Step (3) apply

[1]Work done while the author was an intern at IBM TJ Watson Research Center during Summer 2002.

[2]This was done by retaining only those questions that had 10 words or less, and were not multiple choice.

[3]In these experiments we use HMMNE, a named entity tagger similar to the BBN's Identifinder HMM Tagger (Bikel et al., 1999).

[4]Alta Vista http://www.altavista.com

HMMNE and retain only those sentences that contains at least one of the QTerms plus the ATerm.

For example, we obtain the following sentences for the QA pair "Which country was invaded by the Libyan troops in 1983? - Chad":

1. More than 7,000 Libyan troops entered Chad.
2. An OUA peacekeeping force of 3,500 troops replaced the Libyan forces in the remainder of Chad.
3. In the summer of 1983, GUNT forces launched an offensive against government positions in northern and eastern Chad.

The underlined words indicate the QTerms and the ATerms that helped to select the sentence as a potential way of answering the Question. The algorithm described above was applied to each of the 16,228 QA pairs in our KM database. A total of more than 250K sentences was obtained.

3.2 Sentence Canonicalization

Every sentence obtained from the sentence construction algorithm is canonicalized. Canonicalization of a sentence is performed on the basis of the information provided by HMMNE, the QTerms and the ATerm. Canonicalization in this context may be defined as the generalization of a sentence based on the following process:

1. Apply HMMNE to each sentence obtained from the sentence construction algorithm.
2. Identify the QTerms and ATerm in the answer sentence.
3. Replace the ATerm by the tag "<ANSWER>".
4. Replace each identified Named Entity by the class of entity it represents.
5. If a given Named Entity is also a QTerm, indicate it by the tag "QT".

The following example illustrates canonicalization. Consider the sentence:

More than 7,000 Libyan troops entered Chad.

The application of HMMNE results in:

More than <NUMEX TYPE=CARDINAL>7,000 </NUMEX> <HUMAN TYPE=PEOPLE>Libyan </HUMAN> troops entered <ENAMEX TYPE= COUNTRY>Chad</ENAMEX>.

The canonicalization step gives the sentence:

More than <CARDINAL> <PEOPLE_QT> troops entered <ANSWER>.

3.3 Pattern Extraction

Pattern extraction algorithm.

1. Every sentence obtained from sentence canonicalization algorithm is delimited by the tags "<START>" and "<END>" and then passed through a Suffix Tree. The Suffix Tree algorithm obtains the counts of all sub-strings of the sentence.

2. From the Suffix Tree we obtain only those sub-strings that are at least a trigram, contain both the "<ANSWER>" and the "<QT>" tag and have at least a count of 3 occurrences.

Source	Number of Questions
Trec8	200
Trec9	500
KM	4200

Table 1: Training source and sizes.

Some examples of patterns obtained from the Suffix Tree algorithm are as follows:

1. son of <PERSON_QT> and <ANSWER>
2. of the <ANSWER> <DISEASE_QT>
3. of <ANSWER> at <LOCATION_QT>
4. <ANSWER> was the <ORDINAL> <OCCUPATION _QT> to
5. <ANSWER> was elected <OCCUPATION_QT> of the <LOCATION_QT>
6. <ANSWER> was a prolific <OCCUPATION_QT>
7. <LOCATION_QT> , <ANSWER>
8. <ANSWER> , <LOCATION_QT>
9. <START> <ANSWER> served as <OCCUPATION _QT> from <DATE>
10. <START> <ANSWER> is the <PEOPLE_QT> name for

A set of 22,353 such patterns were obtained by the application of the pattern extraction algorithm from more than 250,000 sentences. Some patterns are very general and applicable to many questions, such as the ones in examples (7) and (8) while others are more specific to a few questions, such as examples (9) and (10). Having obtained these patterns we now can learn the appropriate "weights" to use these patterns in a Question Answering System.

4 Maximum Entropy Training

For these experiments we use the Maximum Entropy formulation (Della Pietra et al., 1995) and model the distribution (Ittycheriah, 2001b),

$$p(c|q,a) = \sum_e p(c|e,q,a)p(e|q,a) \quad (1)$$

The patterns derived above are used as features to model the distribution $p(c|e,q,a)$, which predicts the "correctness" of the configuration of the question, q, the predicted answer tag, e, and the answer candidate, a. The training data for the algorithm consists of TREC-8, TREC-9, and a subset of the KM questions which have been judged to have answers in the TREC corpus[5]. The total number of questions available for training is shown in Table 1.

We perform 3 sets of experiment with different choice of feature sets for training:

1. In the first experiment, the patterns obtained automatically from the web are trained along with the expected type of answer using the Maximum Entropy Framework. We refer to this system as the Pat_Only System. This feature collection consisted

[5]Tagging of answers was done in a semi automatic way by human judges.

	Number of questions correct		
Rank	PAT_ONLY	IBM_TREC11	ME_PAT
1	117	157	167
2	24	21	32
3	16	21	14
4	16	22	11
5	8	8	10
MRR	0.29934	0.37573	0.39703

Table 2: Results on TREC-10.

of roughly 22,353 pattern features along with the 30 different expected answer types (the ones recognized by HMMNE).

2. In the second experiment we use a Statistical QA system that contains bag of words, syntactic and named-entity features. We refer to this system as the IBM_TREC11 System. Details of this system appear in (Ittycheriah and Roukos, 2002). This system has approximately 8,000 features.
3. In the third experiment we add the patterns as additional features to the base system IBM_TREC11 and train the system. We refer to this system as the ME_PAT System. Hence, the total number of features in this system is equal to the sum of the ones in Pat_Only and IBM_TREC11 system.

These systems were trained on TREC-9 and KM and for picking the optimum model we used TREC-8 as held-out test data.

5 Results on TREC-10

We then tested the model on TREC-10. We tabulate the results in Table 2. The TREC-10 collection consisted of 500 questions. The Rank column indicates the number of questions answered by the QA systems with that particular rank. Finally the Mean Rank Reciprocal (MRR) scores are reported.

6 Conclusion and Future Work

Not surprisingly, the PAT_ONLY system shows only average performance as compared to other TREC-10 systems. This is because the system has no information about the question except about its expected answer-type. Hence, the PAT_ONLY system would answer all the questions involving TIME such as: "When was A born?", "When did A die?", "Which year did A start attending college?", "When did A author book B?"with the same answer!

Nonetheless, the ME_PAT results show that surface text patterns are useful for a Question Answering System. Although in these experiments a feature set of 22,353 patterns was trained on approximately 210,000 instances, only 1500 patterns was actually found in the final training data which had a count of at least 8 instances. This suggests that the approach used here to train weights suffers from the problem of having very little training data as compared to the number of features. A much better approach would be to train the weights of the patterns from the unsupervised collection itself. However, the effect of noise introduced due to such unsupervised training is unclear.

The above technique represents a very clean approach to integrating the use of patterns into a QA system. Most of the rule based systems take years to engineer and are very difficult to duplicate. However, a good statistical system can be duplicated to give good performance in a relatively short amount of time.

References

D. Bikel, R. Schwartz and R. Weischedel. 1999. An Algorithm that Learns Whatś in a Name. *Machine Learning Special Issue on NL Learning*, 34, 1–3.

A. Ittycheriah, M. Franz, W. Zhu, A. Ratnaparki and R. Mammone. 2001a. Question Answering Using Maximum Entropy Components Proceedings of the *NAACL* Conference, Pittsburgh, PA, 33–39.

A. Ittycheriah. 2001b. Trainable Question Answering System. *PhD Thesis*, Rutgers, The State University of New Jersey, New Brunswick, NJ.

A. Ittycheriah and S. Roukos. 2002. IBM's Statistical Question Answering System for TREC-11. Proceedings of the *TREC-11 Conference*, NIST, Gaithersburg, MD, 394–401.

KnowledgeMaster. 1999. http://www.greatauk.com. *Academic Hallmarks*.

S. Della Pietra, V. Della Pietra, and J. Lafferty. 1995. Inducing Features of Random Fields. *Technical Report*, Department of Computer Science, Carnegie-Mellon University, CMU-CS-95–144.

D. Ravichandran and E. Hovy. 2002. Surface Text Patterns for a Question Answering system. Proceedings of the *ACL* Conference. Philadelphia, PA, 425–432. 41–47.

M.M. Soubbotin and S.M. Soubbotin. 2001. Patterns of Potential Answer Expressions as Clues to the Right Answer. Proceedings of the *TREC-10* Conference. NIST, Gaithersburg, MD, 134–143.

M.M. Soubbotin and S.M. Soubbotin. 2002. Use of Patterns for detection of likely Answer Strings: A Systematic Approach Answer. Proceedings of the *TREC-2002* Conference. NIST, Gaithersburg, MD, 175–182.