A Factoid Question Answering System Using Answer Pattern Matching

Nagehan Pala Er
Department of Computer Engineering,
Bilkent University, Ankara, Turkey
nagehan@cs.bilkent.edu.tr

Ilyas Cicekli
Department of Computer Engineering,
Hacettepe University, Ankara, Turkey
ilyas@cs.hacettepe.edu.tr

Abstract

In this paper, we describe a Turkish factoid QA system which uses surface level patterns called answer patterns in order to extract the answers from the documents that are retrieved from the web. The answer patterns are learned using five different answer pattern extraction methods with the help of the web. Our novel approach to extract named entity tagged answer patterns and our new confidence factor assignment approach have an important role in the successful performance of our QA system. We also describe a novel query expansion technique to improve the performance. The evaluation results show that the named entity tagging in answer patterns and the query expansion leads to significant performance improvement. The scores of our QA system are comparable with the results of the best factoid QA systems in the literature.

1 Introduction

Question answering is the task of returning a particular piece of information in response to a natural language question. The aim of a question answering system is to present the required information directly, instead of documents containing potentially relevant information. Questions can be divided into five categories (Modolvan et. al., 2002; Schone et. al., 2005): factoid questions, list questions, definition questions, complex questions, and speculative questions. A factoid question has exactly one correct answer which can be extracted from short text segments. The difficulty level of factoid questions is lower than the other categories. In this paper, we present a Turkish factoid question answering system which retrieves the documents that contain answers, and extracts the answers from these retrieved documents with the help of a set of learned answer extraction patterns. List, definition, complex, and speculative questions are out of the scope of this paper.

At TREC-10 QA track (Voorhees, 2001), most of the question answering systems used Natural Language Processing (NLP) tools such as a natural language parser and WordNet (Fellbaum, 1998). However, the best performing system at TREC-10 QA track used only an extensive list of surface patterns (Soubbotin and Soubbotin, 2001). Therefore we have decided to investigate their potential for Turkish factoid question answering. Our factoid question answering system learns answer patterns that are surface level patterns, and it uses them in the extraction of the answers of new questions. Our answer patterns are learned from the web using machine learning approaches. In addition to the creation of raw string answer patterns, we tried different methods for answer pattern creation such as stemming and named entity tagging. Our novel answer pattern creation method using named entity tagging produces most successful results.

One of the important issues in the factoid question answering is the answer ranking. The correct answer for a question should be in the top of the produced answer list by a QA system. The learned answer patterns in our QA system are associated with confidence factors, and their confidence factors indicate their precision values for the training set. The confidence factors of the rules that extracted the answers are also used to rank the answers and this approach produces very good results.

The question answering systems that extract the answers from the retrieved web pages should be able to retrieve the web pages that contain the answers. These QA systems form a search engine query and submit this query to retrieve the related web pages containing the answers. The retrieved web pages may or may not contain the answers. The QA systems can only consider the first retrieved web pages in order to extract an-

swers from them, and the QA systems have a bigger chance to extract the correct answers if the first retrieved web pages contain the answers. In order to increase the chance that the first retrieved web pages contain the possible answers, we apply a novel query expansion approach using answer patterns. Our evaluation results indicate that our query expansion approach improves the performance of our factoid question answering system.

The factoid question answering system described here is the first successful Turkish factoid question answering system. Our new confidence factor assignment approach to the learned answer patterns has an important role in the success of our factoid QA system. The contributions of our paper also include the introduction of a novel query expansion approach and the creation of named entity tagged answer patterns. The performance results of our factoid question answering system are competitive with the results of the state of art QA systems in the literature.

The rest of the paper is organized as follows. Section 2 discusses our answer pattern extraction methods and confidence factor assignments to the extracted answer patterns. In Section 3, we describe the question answering phase of our QA system. Section 4 presents the detailed discussions about the evaluation results. Section 5 contains concluding remarks.

2 Answer Pattern Extraction

In the learning phase of our question answering system, a set of answer patterns are inferred for each question type using the training set of that question type and the web. For each question type, we prepared a set of training examples which consists of question and answer phrase pairs. A query is formed as a conjunction of question and answer phrases in each training example. This query is used to retrieve top documents (*DocSet1*) containing the question and answer phrases. The retrieved documents are used in the extraction of answer patterns without confidence factors. For each training example, we also form a query which only consists of the question phrase. The retrieved documents (*DocSet2*) using this query may or may not contain answer phrases. Two document sets are used in the calculation of the confidence factors of the learned answer patterns.

Although the retrieved documents in *DocSet1* contain both question and answer phrases, question and answer phrases may not appear together in a sentence of a document. In order to determine answer pattern strings, the sentences that contain the question and the answer phrases together are selected from documents, and answer pattern strings are extracted from these sentences. An answer pattern string is a substring that starts with the answer phrase and ends with the question phrase, or starts with the question phrase and ends with the answer phrase. In addition, we extract an answer pattern string with a boundary word in order to determine the boundary of the answer phrase.

After an answer pattern string is extracted, we apply five different methods in order to learn answer patterns: Raw String (*Raw*), Raw String with Answer Type (*RawAT*), Stemmed String (*Stemmed*), Stemmed String with Answer Type (*StemmedAT*), and Named Entity Tagged String (*NETagged*). Raw string methods learn more specific rules than Stemmed string methods, and *NETagged* method learns more general rules.

In order to extract a raw string answer pattern using our *Raw* method, question phrase and answer phrase in an answer pattern string are replaced with appropriate variables QP and AP. QP is replaced with the given question phrase during question answering, and AP is bound to the answer phrase of the question if the pattern matches. The length of the found answer phrase is determined by the boundary word if the answer pattern contains a boundary word. Otherwise, a fixed size is used as its length.

There can be many strings that can match with an answer pattern that is learned using *Raw* method. One reason for this is that there is no type checking for the string to which AP binds. As long as the pattern matches, AP binds with a string. Our *RawAT* method associates AP variables with answer types. An answer type is a named entity type that is determined by our Turkish named entity tagger. During question answering, the found answer phrase is checked by our named entity tagger in order to make sure that it satisfies the type restriction. For this reason, an answer pattern with an answer type is more specific than the corresponding answer pattern without a type.

Answer patterns obtained by raw string methods contain surface level words, and they have to match exactly with words in extracted strings. Stemmed string methods replace words with their stems in answer patterns. In order to match a string with a stemmed answer pattern, all its words are stemmed first and its stemmed version matches with the stemmed answer pattern to ex-

tract the answer. The extracted answer patterns can be still more specific since they may contain specific words. *NETagged* method can further generalize answer patterns by replacing all named entities in the string by typed variables.

After all answer patterns are extracted from the training set, the confidence factors are assigned to these extracted answer patterns. A confidence factor of an answer pattern indicates its accuracy in the training set. In question answering phase, we use only the answer patterns whose confidence factors are above a certain threshold. From two document sets (*DocSet1* and *DocSet2*), the sentences containing the question phrase are collected as a training set for confidence factor assignment. The confidence factor of an answer pattern is the proportion of correct results to all results extracted by that pattern.

3 QA Using Answer Patterns

Our base question answering module uses the given question phrase as a search engine query. Using Bing web search engine top documents containing the given question phrase are retrieved. In these documents, the sentences containing the question phrase are extracted. The question phrases in the retrieved sentences are replaced by QP, and these sentences are used in the answer processing phase.

In the answer processing phase, the answer patterns of the given question type are applied to the selected sentences in order to extract answer phrases. The preprocessing of the sentences may be required depending on the method of the used answer pattern. If the applied method is a raw string method, there is no need for the preprocessing of the sentence, and the raw string answer pattern is directly applied to the sentence. If the answer pattern is a stemmed string answer pattern, all the words are stemmed first, and the answer pattern is applied to the stemmed version of the sentence. If the answer pattern is a NE tagged answer pattern, the sentence is analyzed by the named entity tagger in order to determine all named entities in the sentence, and the answer pattern is applied to the named entity tagged version of the sentence.

If the applied answer pattern matches the sentence, an answer phrase is extracted as a result. If the answer phrase in the applied answer pattern is named entity tagged, the extracted answer phrase must also satisfy conditions of that named entity. The confidence value of an extracted answer is the confidence factor of the matched answer pattern. The top ranked answer is returned as the result of the question.

Our base QA algorithm creates a search engine query and that query only contains the given question phrase. The retrieved documents may be insufficient to extract the correct answer because the query is too general and the retrieved documents may not contain the answer. We want to retrieve documents that contain many sentences holding the question phrase and answer phrase together. Thus, there is a bigger chance that our answer patterns match those sentences, and the correct answer can be extracted. In order to retrieve the documents that are more likely to contain the answer, we use a query expansion approach. The answer patterns with high confidence factors are used to expand the query, so that the more related documents can be retrieved.

4 Evaluation Results

In order to evaluate the performance of our system, we prepared a training set and a test set and they do not contain any common item. Each of them contains 15 question-answer phrase pairs from seven different question types (Author, Capital, DateOfBirth, DateOfDeath, LanguageOfCountry, PlaceOfBirth, PlaceOfDeath). Since we obtained our best results, when we use the answer patterns higher than 0.75 confidence factors, we only used these answer patterns for evaluations. The answer patterns are tested with the question-answer phrase pairs in the test set.

We used four standard evaluation metrics: Precision, Recall, Fmeasure and MRR. Precision is the proportion of the number of correct answers to the number of returned answers, and Recall is the proportion of the number of correct answers to the number of test questions. Fmeasure is the harmonic mean of Precision and Recall. Mean Reciprocal Rank (MRR) considers the rank of the first correct answer in the list of possible answers (Voorhees, 2001).

	MRR	*Recall*	*Precision*	*Fmeas*
Raw	0.28	0.24	0.57	0.34
RawAT	0.31	0.30	0.86	0.44
Stemmed	0.29	0.26	0.57	0.36
StemmedAT	0.30	0.29	0.88	0.44
NETagged	0.45	0.45	0.94	0.61
AllWithNE	0.58	0.56	0.86	0.68

Table 1. Evaluation results

We evaluated each of our five methods separately and their best combination. The evaluation results are given in Table 1. The results in the

columns 2-5 of Table 1 are the average values of the results of the seven question types. The rows 2-6 give the results for individual methods and the last row gives the results of their best combination *AllWithNE* which contains the answer patterns that are learned from methods *RawAT*, *StemmedAT* and *NETagged*. According to the results, our best method is *NETagged* method which accomplishes the best scores for all four evaluation metrics. These results indicate that the usage of named entity tagged string answer patterns increases the performance. The results indicate that the effect of stemming is not as good as expected. The usage of answer types blocks the extraction of most of the incorrect answers.

In our query expansion method, we use the words appearing in the high confidence answer patterns. One way to test the effectiveness of our query expansion mechanism is to measure the change in the number of sentences containing both the question phrase and the answer phrase in the retrieved documents. According to our results, the number of such sentences is increased from 3227 to 6647 when the query expansion is employed. This means that our answer patterns have almost twice the chance to extract answers using query expansion. We applied our answer patterns in our best combination *AllWithNE* to the documents returned as a result of the query expansion. The highest increase (29%) occurred in Recall result because the answers of more questions are retrieved as a result of the query expansion. Precision result is also improved from 0.86 to 0.94 (9% increase). The increase in MMR result is 26% percent and the increase in Fmeasure result is 20%. As a conclusion, the query expansion is a useful tool to improve the performance since it leads to increases in all measures.

QA System	*MRR*
TREC-8 (max,avg,min)	0.66, 025, 0.02
TREC-10 (max,avg,min)	0,68, 0.39, 0.27
Ephyra	0.40
Ravichandran and Hovy's QA sys.	0.57
BayBilmis	0.31
Our Best without query expansion	0.62
Our Best with query expansion	0.73

Table 2. Comparisons of QA systems

Although it is difficult to directly compare the results of our QA system with the published results of other factoid question answering systems, we still discuss the MRR results of our QA system and other factoid question answering systems. Table 2 compares our best MRR results with the MRR results of the other systems (Voorhees, 1998; Voorhees, 2001; Schlaefer and Gieselmann, 2006; Ravichandran and Hovy, 2001; Amasyali and Diri, 2005). Although these scores may not give fair comparisons, they still show that our QA system is competitive to the best factoid QA systems.

5 Conclusion and future work

The answer pattern matching technique has been used successfully for English Factoid QA (Ravichandran and Hovy, 2001; Schlaefer and Gieselmann, 2006; Soubbotin and Soubbotin, 2001), we therefore decided to apply various answer pattern extraction methods for Turkish factoid QA. These methods are compared according to MRR, Fmeasure, Recall and Precision scores. The scores of stemmed string methods are slightly better than the scores of raw string methods, so stemming slightly improves the performance of the system. The scores of *RawAT* and *StemmedAT* methods are better than the scores of *Raw* and *Stemmed* methods, so checking the answer type improves the performance of the system significantly. *NETagged* method has the best scores. So, replacing words with their named entity tags improves the performance.

We have also implemented a novel query expansion approach using answer patterns. We use the most reliable raw string answer patterns to extend queries. The number of sentences containing the answer phrase increases when the query expansion is applied. The performance scores increase significantly when the query expansion is applied.

The question answering system described in this paper is the first successful Turkish factoid question answering system. The evaluation results indicate that the performance of our QA system is comparable with the performances of the state of the art factoid question answering systems.

Investigating the potential of more generic answer patterns is left as a future work. *Stemmed, StemmedAT* and *NETagged* methods extract more generic answer patterns compared to *Raw* and *RawAT* methods and they achieve better results. More generic answer patterns can be extracted by using linguistic techniques such as phrase chunking and morphological analysis. We believe that combining different answer processing techniques can improve the performance of the QA system significantly.

References

M.F. Amasyalı and B. Diri, Bir soru cevaplama sistemi: Baybilmiş, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 2005 (in Turkish).

C. Fellbaum, *WordNet: An Electronic Lexical Database*, The MIT Press, (1998).

D. Modolvan, M. Pasca, S. Harabagiu and M. Surdeanu, Performance issues and error analysis in an open-domain question answering system, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)* (2002).

R. Ravichandran and E. Hovy, Learning surface text patterns for a question answering system, *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (2001).

N. Schlaefer and P. Gieselmann, A pattern learning approach to question answering within the Ephyra framework, *Proceedings of the 9th International Conference on Text, Speech and Dialogue* (2006).

P. Schone, G. Ciany, R. Cutts, J. Mayfield and T. Smith, QACTIS-based question answering at TREC-2005, *Proceedings of the 14th Text Retrieval Conference* (2005).

M. Soubbotin and S. Soubbotin, Patterns of potential answer expressions as clues to the right answer, *Proceedings of the 10th Text Retrieval Conference* (2001).

E.M. Voorhees, The TREC-8 question answering track report, *Proceedings of the 8th Text Retrieval Conference* (1999).

E.M. Voorhees, Overview of the TREC 2001 question answering track, *Proceedings of the 10th Text Retrieval Conference* (2001).