

A Pattern Learning Approach to Question Answering within the Ephyra Framework

Nico Schlaefel¹, Petra Gieselmann¹, Thomas Schaaf², and Alex Waibel^{1,2}

¹ Interactive Systems Labs, ITI, Universität Karlsruhe
Am Fasanengarten 5, 76131 Karlsruhe, Germany
{nico, petra}@ira.uka.de

² Interactive Systems Labs, Carnegie Mellon University
407 S. Craig Street, Pittsburgh, PA 15213
{tschaaf, waibel}@cs.cmu.edu

Abstract. This paper describes the Ephyra question answering engine, a modular and extensible framework that allows to integrate multiple approaches to question answering in one system. Our framework can be adapted to languages other than English by replacing language-specific components. It supports the two major approaches to question answering, knowledge annotation and knowledge mining. Ephyra uses the web as a data resource, but could also work with smaller corpora. In addition, we propose a novel approach to question interpretation which abstracts from the original formulation of the question. Text patterns are used to interpret a question and to extract answers from text snippets. Our system automatically learns the patterns for answer extraction, using question-answer pairs as training data. Experimental results revealed the potential of this approach.

1 Introduction

Question answering is a form of information retrieval that deals with natural language questions. The main goal is to retrieve explicit answers to questions rather than whole documents. This task has already been addressed by various systems following both linguistic and statistical approaches and has been intensively investigated in the Text REtrieval Conference (TREC) [1].

In this paper, we describe the Ephyra framework for open-domain question answering which allows to efficiently implement and combine new techniques for question analysis and answer extraction. Ephyra can easily be integrated in different NLP applications, such as dialog management, and it can be adapted to other languages. As part of this framework, we developed a pattern matching approach. Ephyra automatically learns text patterns that can be applied to text passages for answer extraction. The system is trained on question-answer pairs and uses conventional web search engines to fetch text snippets suitable for pattern extraction. A second set of patterns is used to interpret questions. The interpretation of a question abstracts from its original formulation but yet preserves its semantics. The pattern matching approach has a high precision but a relatively low recall. Therefore, we combined it with a number of backup techniques for query formation and answer selection that are used when this approach fails. Since the focus of this paper is on the pattern matching approach, we do not discuss the backup techniques in detail.

Section 2 gives an overview of related work on question answering and pattern learning. In Section 3, we introduce the Ephyra framework. Section 4 is about our pattern matching approach. Section 5 gives experimental details and results of the pattern matching approach combined with backup techniques. Section 6 is a conclusion and an outlook on future work.

2 Related Work

Many of today's question answering systems, such as AnswerBus [2] or AskMSR [3], use a common architecture that comprises the following three modules: query formation, search and answer selection. Ephyra also consists of these three modules on the top level and some of its backup techniques (e.g. answer type testing and query reformulation) are based on these systems.

Lin et al. introduced the idea of combining knowledge annotation and knowledge mining techniques in a single framework [4]. Knowledge annotation allows to extract answers for frequent question classes from semi-structured sources such as web sites or web services, while knowledge mining applies statistical methods to unstructured sources such as the entire web. Ephyra supports these two paradigms and adds learning techniques on top of that.

The LAMP question answering system [5] follows a pattern mining and matching approach. It uses manually created text patterns to classify a question and to extract a key phrase, which is an object or event that the questions asks about. A second set of patterns is learned automatically and is used to extract answers from search results. The approach is limited to a single key phrase, and thus questions such as "How many calories are there in a Big Mac?" (cf. TREC8, question 56) cannot be handled because of the two key phrases "calories" and "Big Mac". Another shortcoming of LAMP is the relatively small set of 22 question classes. We introduce a more general approach to question interpretation that supports multiple key phrases and that distinguishes more classes to ensure that the right property of the key object is retrieved.

3 The Ephyra Framework

Ephyra is a modular framework that can easily be extended by integrating additional techniques. The system is also adaptable to other languages, which was achieved by separating language-specific from language-independent code and by defining patterns that are specific for the English language in separate resource files. The overall architecture is shown in Figure 1.

The question normalization component applies a number of normalization steps to the question string. It drops punctuation and quotation marks and modifies verb constructions in order to ensure that the question meets the premises of the following components.

A query generator transforms the question string into one or more queries. Ephyra comprises a query generator that builds a simple "bag of words" and a more sophisticated one that rephrases the question, anticipating the format of an answer. The question interpreter is part of our pattern learning approach (see Section 4.1). Our framework can be extended by simply plugging in additional query generators.

The queries are handed over to the search module, which comprises two types of searchers. Knowledge miners use conventional IR systems to query unstructured sources (e.g. Google

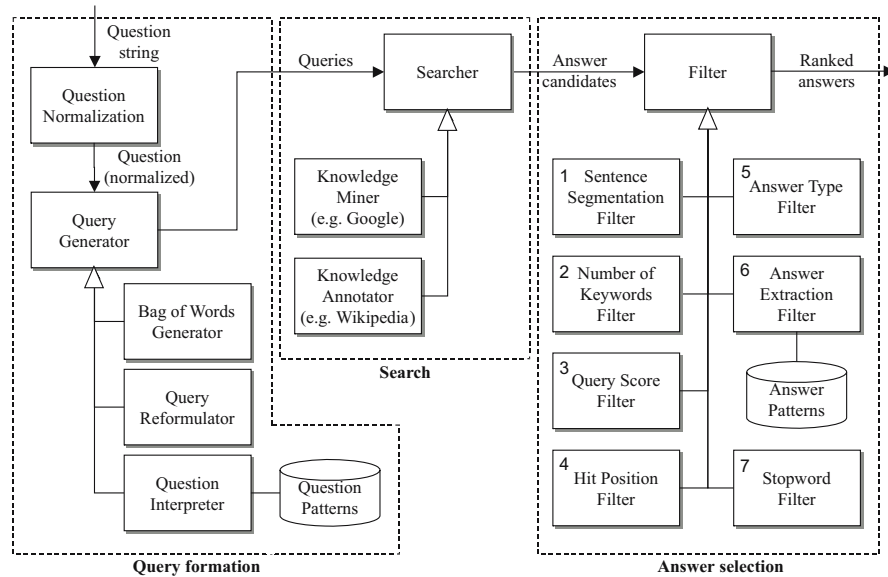


Fig. 1. Overview of the Ephyra Architecture

to search the web). Knowledge annotators allow to integrate resources such as web sites or web services that provide semi-structured information (e.g. the CIA World Factbook). To incorporate an additional knowledge source, one can simply create a new searcher.

Finally, the search results are processed by a set of filters. A filter usually checks the results for a specific feature and promotes those results that are promising with respect to that feature. For instance, the answer type filter promotes answers that are of the expected type (such as *date* or *location*). In addition, a filter can create new results from the existing ones (e.g. by breaking down a text snippet into sentences) and it can drop candidates that are unlikely to answer the question. The answer extraction filter also belongs to our pattern learning approach (cf. Section 4.2). By adding an additional filter, a new feature for answer evaluation or extraction can be integrated.

4 A Pattern Learning Approach

The Ephyra system uses two types of text patterns in the query formation and answer selection components described previously.

- *Question patterns* are applied to question strings to interpret the questions and to transform them into queries.
- *Answer patterns* are used to extract answers from relevant text snippets and to rank them.

While the question patterns need to be specified manually, Ephyra can automatically learn answer patterns, using question-answer pairs as training data.

4.1 Question Interpretation

Our pattern learning approach is based on the assumption that each question can be reduced to three components: A question asks for a *property* of a *target* in a specific *context*. For example, the question "How many calories are there in a Big Mac?" can be interpreted as follows:

- Property: NUMBER
- Target: "calories"
- Context: "Big Mac"

The question asks for a property of the target object "calories", which is its number. The context object "Big Mac" narrows down the scope to a particular food. Together, these three components form the interpretation of the question.

Ephyra knows about 70 properties such as the DATE of an event, the NAME of a person or organization and the LONGFORM of an abbreviation. Each property is associated with a number of frequent question patterns. These patterns are basically regular expressions but in addition, they contain exactly one target tag <T>, indicating the target of the question, and 0 to *n* context tags <C>, indicating context information. The tags can be combined with *object types* to restrain the format of the objects they represent (e.g. <T_ABBR> means that the target object must be an abbreviation).

Ephyra interprets a question by sequentially applying all the question patterns. If the question matches a pattern, Ephyra knows the property the question asks for and the substrings of the question corresponding to the <T> and <C> tags are extracted as target and context objects. In our example, the question could be interpreted with a pattern "how many <T> are (there)?(in|on|at) <C>" belonging to the property NUMBER.

The classification of questions based on properties is reasonable, since it ensures that there are just enough classes to retrieve the right aspect of a target. Obviously, it is not feasible to cover all properties that a question could possibly ask for, but this is also not necessary because a question can usually be interpreted in more than one way. For instance, the question "What is the name of the wife of Bill Clinton?" can be interpreted as follows:

- Property: WIFE, Target: "Bill Clinton"
- Property: NAME, Target: "wife of Bill Clinton"

The first interpretation is the preferable one because the target is simple and it supports synonyms such as "spouse". But even without a property WIFE, the question can still be handled with the more general property NAME.

The interpretations are transformed into search engine queries. The query string is the concatenation of the target object and all context objects in quotation marks plus additional keywords in the question. In our example, Ephyra would build the string "calories" "Big Mac". The query strings are then used to fetch text snippets from the Google and Yahoo search engines.

4.2 Answer Extraction

Each of the properties is also associated with a set of answer patterns. These patterns are used to extract answer candidates from text snippets that contain the target and context objects of

the question. The format of the answer patterns is similar to the question patterns described in the previous section. An answer pattern contains a target tag <T>, an arbitrary number of context tags <C> and a property tag <P>. In addition, each pattern is assigned a confidence value that estimates the reliability of answers extracted with that pattern.

Ephyra replaces all occurrences of target or context objects in a text snippet by the respective tags (<T> or <C>) and successively applies the answer patterns for the particular property to the text snippets. Whenever a snippet matches a pattern, the part of the snippet that corresponds to the <P> tag is extracted. In our example, the snippet "One Big Mac contains 560 calories and 32 grams of fat" is transformed into "One <C> contains 560 <T> and 32 grams of fat". The pattern "contains <P> <T>" can then be used to extract the property "560". If the extracted string does not occur in the set of answer candidates found so far, a new candidate is created and its initial score is set to the confidence measure of the pattern. Else, the score of the existing candidate is simply incremented by that confidence value. This mechanism results in a ranked list of answer candidates with more promising candidates coming first.

4.3 Pattern Learning

Initially, the properties and question patterns need to be specified manually. Ephyra can then automatically learn answer patterns, using question-answer pairs as training data: At first, the answer patterns are extracted from text snippets. Secondly, the patterns are assessed and patterns that are unreliable or too specific are dropped.

Properties and question patterns. We determined the properties and question patterns by analyzing the questions from the TREC9 question answering track (700 questions). Firstly, we derived simple patterns by replacing the target and context objects in the questions with the respective tags. Then we generalized the question patterns by adding synonyms and alternative formulations. To avoid ambiguous patterns, we associated some of the <T> and <C> tags with *object types* to put restraints on the target and context objects (cf. Section 4.1). For instance, these are some question patterns for the property NUMBER:

```
how many <T> are there
how many <T> are (there )?(at|in|on) <C>
what is the number of <T> (at|in|on) <C>
```

Extraction of answer patterns. The Ephyra system can be trained on question-answer pairs. We used the questions and answers from the TREC9 QA track, the same training set used to identify the properties and question patterns. At first, the questions in the training set are interpreted by applying the question patterns. Questions that cannot be interpreted are discarded. For each interpretation, a tuple consisting of a target, an arbitrary number of context objects and the answer (i.e. the property) is generated. It is possible to selectively add tuples for properties that are not sufficiently covered by the training data.

Ephyra then uses the tuples to query the Google and Yahoo search engines and fetches the text snippets from the search results. In our running example, the query string is: "calories" "Big Mac" "560". The answer is included in the query because we are interested in snippets that contain both the target and the property. All occurrences of target,

context and property objects in the snippets are replaced by the tags <T>, <C> and <P> respectively. For example, the snippet "One Big Mac contains 560 calories and 31 grams of fat" is transformed into "One <C> contains <P> <T> and 31 grams of fat". Now the answer patterns can be extracted from the snippets. This is done by applying the following two regular expressions (similar to the patterns used in [5]):

- $\backslash\text{B}\langle\text{T}\rangle\backslash\text{B}(.*)\backslash\text{B}\langle\text{P}\rangle\backslash\text{B}\backslash\text{s}*(\backslash\text{W}|\backslash\text{w}+)$
- $(\backslash\text{W}|\backslash\text{w}+)\backslash\text{s}*\backslash\text{B}\langle\text{P}\rangle\backslash\text{B}(.*)\backslash\text{B}\langle\text{T}\rangle\backslash\text{B}$

An answer pattern is basically a regular expression that covers a target tag <T>, a property tag <P> and any characters in between these tags. In addition, it covers one word or special character preceding or following the <P> tag (depending on whether the <P> tag comes before or after the <T> tag). In our example, the second of the above regular expressions would extract the answer pattern "contains <P> <T>".

Assessment of answer patterns. Since this approach can result in a huge amount of patterns and thus in a poor response time, Ephyra assesses the answer patterns and drops patterns that are too specific or unreliable. Again, the starting point is a set of question-answer pairs (we reused the TREC9 data). Similarly to the previous step, Ephyra interprets the questions and generates query strings from the interpretations and answers. It then queries the search engines, fetches the snippets and replaces all occurrences of target and context objects by the <T> and <C> tags.

The algorithm then applies each of the previously learned answer patterns to all the snippets for the respective property and judges the extracted answers. For each answer pattern a , the algorithm records how often it could be used to extract a correct answer ($correct_a$) and how often it extracted a wrong answer ($incorrect_a$). Furthermore, for each property p , the total number of snippets is recorded ($snippets_p$). These values are used to compute the following two measures:

$$Confidence_a = \frac{correct_a}{correct_a + incorrect_a}, \quad Support_a = \frac{correct_a}{snippets_p}$$

These measures are then compared to thresholds. Patterns with a low confidence are considered to be unreliable and are dropped. A low support value means that a pattern is too specific and thus it is also dropped. We chose the thresholds to represent a good compromise between response time, precision and recall. The confidence values are reused to judge answers (see Section 4.2).

5 Experimental Details and Results

We evaluated Ephyra on the 200 TREC8 questions. For each question, our system returned a ranked list of up to five answers. These answers were judged manually by means of the patterns provided by TREC. Table 1 summarizes the results. Our system first tried to answer a question using knowledge annotation, then pattern matching, and if this approach failed it tried the backup techniques. The precision is the percentage of questions with at least one correct answer. The MRR (Mean Reciprocal Rank), as used in the TREC8 evaluation [6], takes the rank of the first correct answer into account.

Table 1. Performance of Ephyra on the TREC8 questions

	Questions	# Correct	Precision	MRR
Knowledge annotation	4	3	0.75	0.75
Pattern learning	96	55	0.57	0.40
Backup	99	47	0.47	0.32
All	200	105	0.53	0.36

The results are encouraging, showing that the pattern learning approach has a high precision, but on the other hand a relatively low recall. Therefore, the combination with backup techniques is reasonable. The confidence scores of the answers are good indicators for their reliability and can be used to merge them with answer candidates from other approaches. The pattern learning approach also has the benefit of mainly returning exact answers instead of sentences or sentence fragments. We only used about 700 questions to derive the properties and to generate the question and answer patterns and thus we assume that the performance of our approach can be improved substantially by increasing the amount and quality of the training data.

6 Conclusion and Outlook

We introduced the Ephyra question answering system, a modular framework that allows to combine multiple approaches to question answering in a single system. Ephyra can be integrated in other NLP applications and is adaptable to languages other than English. Based on this framework, we developed a new approach using text patterns to interpret questions and to extract answers from text snippets. The patterns for answer extraction are learned automatically. We find the concept of the interpretation of a question (comprising a target, context information and a property) promising. The interpretation is largely formulation-independent and thus questions and answers may use different terms. In contrast to previous approaches, multiple key phrases within a question are supported. Furthermore, the classification of questions based on properties results in just enough classes to ensure that the right aspect of a target is retrieved.

However, the answer patterns used by our approach are quite specific and thus require a redundant source. Currently, we are exploring how more sophisticated linguistic techniques such as phrase chunking and named entity tagging can be deployed to derive more generic patterns from the patterns returned by our learning algorithm. We believe that in this way the recall can be improved significantly and at the same time the number of patterns can be reduced. These NLP techniques could also be used to extract the target and context objects from a question, which would allow to automatically learn additional question patterns.

Acknowledgement

This work was supported in part by the DFG as part of the SFB 588 and by the European Commission under project CHIL (contract #506909).

References

1. Voorhees, E., Buckland, L.: The 14th Text REtrieval Conference Proceedings (2005).
2. Zheng, Z.: Answerbus question answering system. Proceedings of the Human Language Technology Conference (2002).
3. Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A.: Data-intensive question answering. Proceedings of the 10th Text REtrieval Conference (2001).
4. Lin, J., Fernandes, A., Katz, B., Marton, G., Tellex, S.: Extracting answers from the web using knowledge annotation and knowledge mining techniques. Proceedings of the 11th Text REtrieval Conference (2002).
5. Zhang, D., Lee, W.: Web based pattern mining and matching approach to question answering. Proceedings of the 11th Text REtrieval Conference (2002).
6. Voorhees, E., Tice, D.: The TREC-8 question answering track evaluation. Proceedings of the 8th Text REtrieval Conference (1999).