QA Extension for Xtrieval: Contribution to the QAst track

Jens Kürsten, Holger Kundisch and Maximilian Eibl
Chemnitz University of Technology
Faculty of Computer Science, Dept. Computer Science and Media
09107 Chemnitz, Germany
[jens.kuersten | holger.kundisch | maximilian.eibl] at cs.tu-chemnitz.de

Abstract

This article describes our first participation at the *QAst task* of the CLEF campaign 2008. We submitted 4 experiments in total, two for each subtask t1 and t4. These subtasks employed manual speech transcription collections. We used the *Stanford Named Entity Recognizer* for tagging named entities and the *CRFTagger - Conditional Random Fields Part-of-Speech (POS) Tagger for English*. The passage retrieval was done with the *Xtrieval* framework and its Apache Lucene implementation. For the classification of the question hand-crafted patterns were implemented. Our experiments achieved an accuracy of about 20%. The rate of returned NIL answers was too high for all of our experiments.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Manual Speech Transcripts

1 Introduction

This article describes the design, architecture and evaluation of a prototype extension of the *Xtrieval* framework [2]. It is used for the participation at the *QAst task* of the CLEF 2008 campaign. The task concerns with two main types of questions: factual questions (ca. 70%) and definition questions (ca. 20%). The remaining proportion of 10% contains questions that could not be answered by querying the provided collections. The organizers of the task defined 10 types of named entities and they divided the definitional questions into 4 subsets (see *QAst task* guideline[1]).

We developed our prototype with respect to the restrictions and assumptions of the task. The prototype includes all classical components of a QA system: (a) question classification, (b) passage retrieval, (c) answer extraction and (d) natural language processing (NLP) components for named entity recognition (NER) and a part-of-speech (POS) tagger. Since we are newcomers to the topic of QA, we decided to start with two simpler tasks (t1 and t4) on manual speech transcriptions. These collections allow a rather simpler strategy

[1] http://www.lsi.upc.edu/~qast/2008

for the identification of the right answer to a factual or definitional question, because no phonetic analysis is necessary.

The remainder of the article is organized as follows. In section 2 we describe our system and its architecture. Section 3 shows the results of our submitted experiments. A summary of the result analysis is given in section 4. The final section concludes our experiments with respect to our expectations and gives and outlook to future work.

2 System Architecture

The main architecture of our prototype is illustrated in figure 1. We use the *Xtrieval* framework with Apache Lucene[2] as core implementation to retrieve answer candidates from our passage index. Both questions and collections are preprocessed with two kinds of taggers. For the selection of the final answer(s) we use: (a) the RSV's of the returned passages, (b) the class the current question belongs to and (c) the NE and POS tags from question and collection.

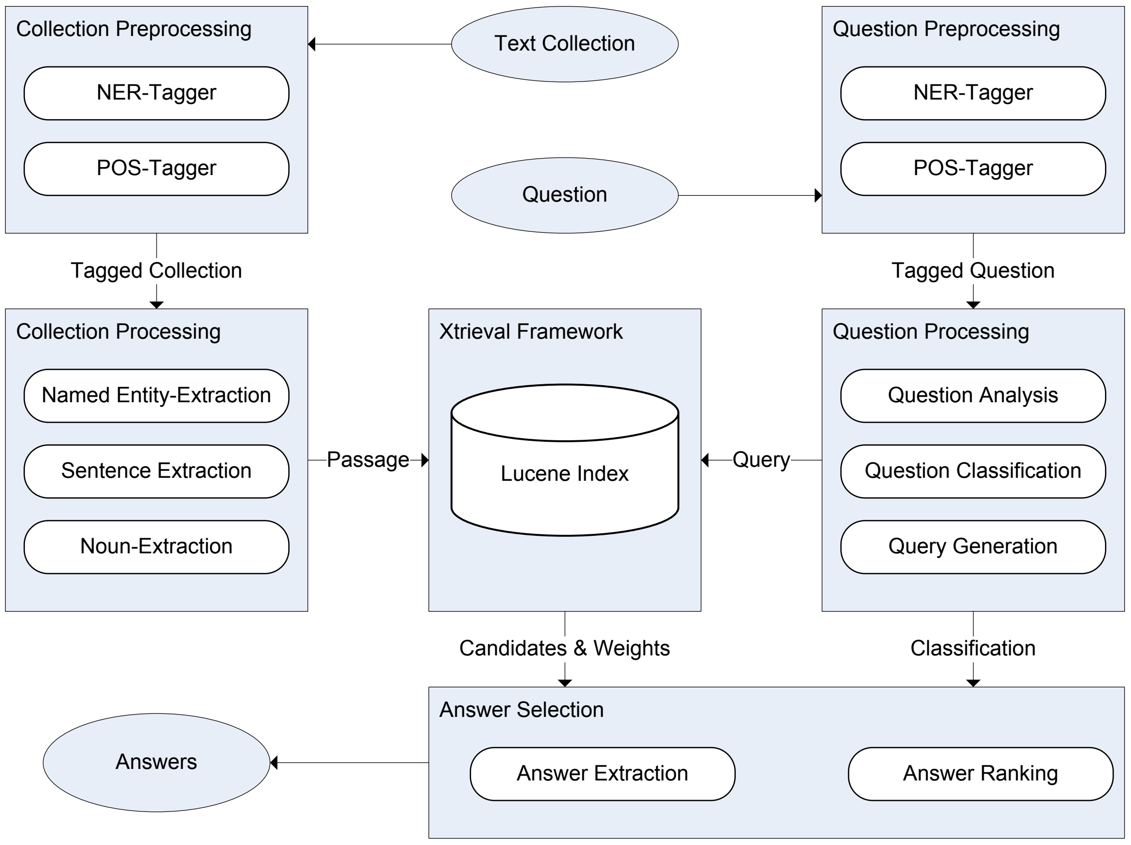

Figure 1: General System Architecture

2.1 Collection and Question Preprocessing

A NLP component could be beneficial for running experiments for the *QAst task* with factual and definitional questions, because with the help of this tagger some answers to the question could be identified directly by

[2]http://lucene.apache.org
[2]http://lucene.apache.org

searching the collection for specific tags. Since many NLP systems exist in the community and also because we are not experts in the NLP topic, we decided to compare different systems. We also investigated whether the systems were adaptable to our special needs for the task. For a more robust QA system we thought it might be useful to integrate both a NER and a POS tagger. The prerequisite for both systems was Java compatibility, because our prototype and the *Xtrieval* framework are implemented in Java. We did not bother when the support for Java was provided by a Java API. In the two following subsections we go into the details of the tagging systems we applied in this work.

2.1.1 NER Tagger

In our system we used the *Stanford Named Entity Recognizer*[3] (SNER) for named entity recognition. It is used for fast extraction of answers to questions, that imply a certain type of named entity [1]. We chose the SNER tagger, because it contains predefined classifiers that are very similar to the entities that are defined in the task description. Unfortunately, it does not cover all the defined entities, but it supports three types: (a) *person (PER)*, (b) *organization (ORG)* and (c) *location (LOC)*. Additionally, the SNER tagger has another classifier called *MISC* for entities that could not be assigned to a specific class. In some preliminary test the SNER tagger achieved quite formidable recognition rates. Table 1 illustrates the statistics of named entities that were assigned with the SNER tagger. It shows the distribution of the recognized entities and the rate (NE2TR) of all recognized tags and the number of unique terms (UT) for the two collections.

Table 1: Distribution of detected named entities

corpus	*# PER*	*# ORG*	*# LOC*	*# MISC*	*# UT*	*NE2TR*
CHIL-manual (t1)	52	71	40	47	2264	09.28%
EPPS-manual (t4)	47	68	69	35	1418	15.28%

2.1.2 POS Tagger

Besides the NER tagger, a POS tagger is also a vital component in our system. This is due to the fact that almost every step depends on the extraction of particular lexical categories. During the query formulation procedure nouns are extracted or during search for measures we look for adjectives just to give a few examples. We use the *CRFTagger - Conditional Random Fields Part-of-Speech (POS) Tagger for English* [3] for POS tagging. It is free, implemented in Java and achieves a high recognition rate[4]. A comparison to the *Stanford Log-linear Part-Of-Speech Tagger*[5] [5],[4] on the test collections showed marginal better performance of the CRFTagger.

2.2 Question Classification

In this step the question is analyzed to decide whether the question is a factual or a definitional one. Therefore we simply apply a hand-crafted pattern to the interrogative. If the question starts with what/who is/are we regard the question as definitional question. Otherwise we assume the current question is a factual question.

2.3 Question Analysis

The types of the named entities are also very important for the analysis of the question itself. The question could be classified by using the 10 types of entities defined in the task guideline. Due to that fact, we try to assign at least one type of entities to the question by using hand-crafted patterns that contain special words. Again to give an easy example one might imagine that a question starting with *how many/much/long/old* is most likely to be answered with a *measure*. We defined a number of these patterns to help estimating what

[3]http://nlp.stanford.edu/software/CRF-NER.shtml
[4]http://crftagger.sourceforge.net/
[5]http://nlp.stanford.edu/software/tagger.shtml

might be a right answer to the question. There are a number of easy types of entities in the group of the 10 entities proposed in the guideline, like numbers or locations. But there are also a number of hard entities like color and shape. When our system is not able to assign a single class to a question, it assigns a number of classes.

2.4 Passage Extraction

We separated the test collection into phrase for the retrieval stage. For the separation of the collection we simply use punctuation marks and all sentences were indexed with the *Xtrieval* framework. We assume this is a good solution because if the passages are larger than it might be much harder to extract the correct answer. Contrary, when we had used smaller passages it may happen that a correct answer is split into two different passages.

2.5 Answer Finding Procedure

The answer finding procedure mainly consists of four important step. At first a query is formulated and feed into the retrieval system. The retrieval system ranks the passages based on their RSV and returns the top passages. Thereafter, the best matching passage is selected and the corresponding answer will be extracted. In the next subsections we go into the details of these four steps.

2.5.1 Query Formulation Procedure

At first, we need to formulate a query to the IR system. The query has to be created depending on the type and the content of the question. Therefore, the system extracts nouns with the help of the POS tagger and assigns weights to the different types of part of speech, e.g. proper names/nouns (NNP/NNPS), which have to occur in a corresponding passage or a standard noun (NN/NNS) that could occur in the relevant passage. The system automatically forms phrase queries for neighboring nouns.

2.5.2 Passage Ranking

The ranking of the passages is defined by the RSV for the passage returned from the IR system. There is only one exception: when the second method for the answer extraction is used (see corresponding section below) the score will be the inverse of the calculated distance value.

2.5.3 Passage Selection

The formulated query is fed into the IR system, which queries the passage index. If this step does not return any relevant passages, a fall-back algorithm will be applied. This method simplifies the query by removing possible phrases or changing a mandatory term into an optional term. As a matter of principle we do only consider the first passage returned by the IR system, because we observed that in the test set the best matching passage contained the correct answer in many cases. This could help us to achieve a high precision, which is very important in a QA scenario from our point of view.

2.5.4 Answer Extraction

The module for the answer extraction consists of two parts. At first, we try to find corresponding answers for special classes of questions by using the POS or NER tags. For example, if the question asks for values or measures, the system will search for adjectives and tries to combine groups of words to build a complete answer like *more than five hundred*. If a posed question contains a person, organization or location, the system will use the terms that were tagged by the SNER tagger. Unfortunately, we discovered that quite a large number of entities are tagged wrong, i.e. they are tagged with the wrong class. For example persons can be recognized as organizations and vice versa. Therefore we think we should use an NER tagger that was adapted or trained with the collections we use for the experiments.

If the first part of the answer extraction did not return any answers or the question was not assigned to a single type a fall-back strategy will be applied. We analyzed some experimental answers on the development data and we observed that the correct answers to some questions are near the terms that occurred in the question. Therefore, we implemented an algorithm to take advantage of this observation. It calculates the distances between all nouns in a sentence that do also occur in the query. We use this measure for the fall-back and return the noun with the lowest overall distance to all query terms. This calculation was only used when the best matching passage returned by the IR system had a RSV that was higher than a certain threshold. This threshold is necessary because otherwise the system would return an answer for every question even for those that are not supported by the collection.

3 Experimental Results

The general setup of the system was discussed in the preceding sections. We submitted 4 experiments in total. For both collections (CHIL and EPPS) we tested a configuration (cut1*) that returned only one answer per question. We also submitted another configuration (cut2*), where up to 3 answers per questions were returned. Tables 2 and 3 present the results of the evaluation.

Table 2: QAst evaluation results

run ID	*corpus*	*# Q*	*# CA*	*# MRR*	*ACC*
cut1_t1a	CHIL-manual	100	16	0.16	16.0%
cut2_t1a	CHIL-Manual	100	24	0.20	17.0%
cut1_t4a	EPPS-manual	100	21	0.21	21.0%
cut2_t4a	EPPS-Manual	100	23	0.22	21.0%

In table 2 we have the following values: (Q) the total number of question that were processed, (CA) the total count of correct answers, (MRR) the mean reciprocal rank and (ACC) the overall accuracy of the system configuration. The results show that the proposed prototype did only answer a fifth of all questions correctly. Interestingly, the accuracy of the first answer is very high. Another observation can be made by looking at the results for the two different collections. We found a higher number of correct answers for the considerably smaller EPPS collection, which might be due to the larger ratio of recognized named entities. But another reason could also be the distribution of the question types over both collections.

Table 3: QAst evaluation results in detail

run ID	*# 1Q*	*# 2Q*	*# 3Q*	*# MQ*	*# R*	*# X*	*# U*	*# W*	*# NIL*	*NIL=R*
cut1_t1a	100	0	0	1.0	16	1	0	83	53	3
cut2_t1a	68	17	15	1.5	17	1	0	82	60	5
cut1_t4a	100	0	0	1.0	21	1	0	78	41	6
cut2_t4a	64	20	16	1.6	21	1	0	78	44	6

Table 3 goes into the details of the evaluation. It shows the following values: ([1|2|3|M]Q) the total number of answers per question and the total number of all assessments: correct (R), inexact (X), unsupported (U), wrong (W), NIL answers (NIL) and correct NIL answers (NIL=R). The analysis of the result shows, that the accuracy of the system is acceptable, but the total number of correct answers is much too low. Generally the total number of NIL answers is too large.

4 Result Analysis - Summary

The following list provides a summary of the analysis of our retrieval experiments for the *QAst task* at CLEF 2008:

- *Task 1a (CHIL-manual)*: Both experimental configurations of the system achieved an accuracy slightly below 20%. Interestingly, this accuracy was achieved by generating only one answer per question (experiments cut1*). The total number of NIL answers is above 50%, which is an almost inacceptable value.

- *Task 4a (EPPS-manual)*: Both experimental configurations of the system achieved an accuracy slightly above 20%. The total number of NIL answers is above 40%, which is also an almost inacceptable value.

5 Conclusion and Future Work

The experiments for the CLEF 2008 *QAst task* allow us to draw some conclusions on how to improve the quality of the used system. At first, we should improve the ratio of correctly recognized named entities either by training the tagger for the special collection or by using a more general tagger. With this improvement we could probably assign more correct classes to the questions posed, which will boost the total number of correct answers without deteriorating the accuracy of the system. Additionally, we could improve or vary the passage extraction procedure to improve the passage retrieval itself. Last but not least one could try to implement an answer extraction strategy that is adapted to find more answers to raise the total number of answered question. This could easily be achieved by tuning the RSV based parameter in the answer extraction fall-back strategy to lower values.

References

[1] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, June 2005.

[2] Jens Kürsten, Thomas Wilhelm, and Maximilian Eibl. Extensible retrieval and evaluation framework: Xtrieval. *LWA 2008: Lernen - Wissen - Adaption, Würzburg, October 2008, Workshop Proceedings*, October 2008, to appear.

[3] Xuan-Hieu Phan. Crftagger: Crf english pos tagger. 2006.

[4] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of HLT-NAACL 2003*, pages 252–259, 2003.

[5] Kristina Toutanova and Christopher Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70, 2000.