DFKI–LT at QAST 2007: Adapting QA components to mine answers in speech transcripts

Günter Neumann and Rui Wang
LT lab, DFKI, Saarbrücken
neumann|wang.rui@dfki.de

Abstract

The paper describes QAst-v1 a robust question answering system for answering factoid questions in manual and automatic transcriptions of speech. Our system is an adaptation of our text–based crosslingual open–domain QA system that we used for the Clef main tasks. In particular we assume that good answer candidates to factoid questions are named entities which are type–compatible with the expected answer type of the question. The main features of QAst-v1 are: use of preemptive off-line annotation of speech transcripts with sentence boundaries, chunk structures and named entities (NEs); construction of a fulltext search index using words and all found NEs; use of robust Wh-analysis component to determine shallow dependency structures, recognition of NEs, and expected answer type (EAT); use of EAT–driven retrieval of sentences and answer candidates; use of redundancy as an indicator of good answer candidates. The main focus of our effort was on the technical realization of a first QAST research prototype making use of as many of our existing QA components as possible. The results of evaluating the system's performance by QAST 2007 were as follows: for subtask T1 (Question-Answering in manual transcriptions of lectures) we achieved an overall accuracy (ACC) of 15% and a mean reciprocal rank (MRR) of 0.17; for subtask T2 (Question-Answering in automatic transcriptions of lectures) we obtained 9% (ACC) and 0.09 (MRR).

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.2.3 [**Database Managment**]: Languages—*Query Languages*; I.2 [**Artificial Intelligence**]: I.2.7 Natural Language Processing

General Terms

Measurement, Performance, Experimentation

Keywords

QAst pilot task, Question answering, speech transcripts from lectures

1 Introduction

The focus of the new Question Answering on Speech Transcripts (QAst) track within CLEF 2007 is on extracting answers to written factoid questions in manual and automatic transcripts of records of spoken lectures and meetings. Although the basic functionality of a QAst–based system is similar to that of a textual QA–system the nature of the different scenarios and answer sources provoke new challenges.

The answer sources for Clef and Trec–like systems are usually text documents like news articles or articles from Wikipedia. In general, an article of such a corpora describes a single topic using a linguistically and stylistically well–formed short text which has been created through a number of revision loops. In this sense, such an article can be considered as being created off–line for the prospective reader. By contrast, transcripts from lectures or meetings are live records of spontaneous speech produced incrementally (or on–line) in human–human interactions. Here, revisions (of errors or refinements) of utterances take place explicitly and immediately or not at all. Thus, speech transcripts also have to encode such properties of incremental language production, like word repetition, error corrections, refinements or interruptions. Consequently, transcripts are less well–formed, stylistic and fluent as written texts. Furthermore, in case of automatic transcripts errors and language gaps caused by the used automatic speech recognition system also make things not easier for a QAst–based system (see also the Background tap of the QAst Clef 2007 web page at http://www.lsi.upc.edu/∼qast/index.html). It seems that QA on speech transcripts demands a high degree of robustness and flexibility from the QA components and its architecture.

Nevertheless, the component architecture of a QAst–based system is similar to that of a textual QA–system and consists of the following core functionality: NL question analysis, retrieval of relevant snippets from speech transcripts, answer extraction, and answer selection. Therefore, we decided to develop our initial prototype QAst-v1 following the same underlying design principles that we used for our textual QA system and by the adaptation of some of its core components, cf. [3, 5].

2 System Overview

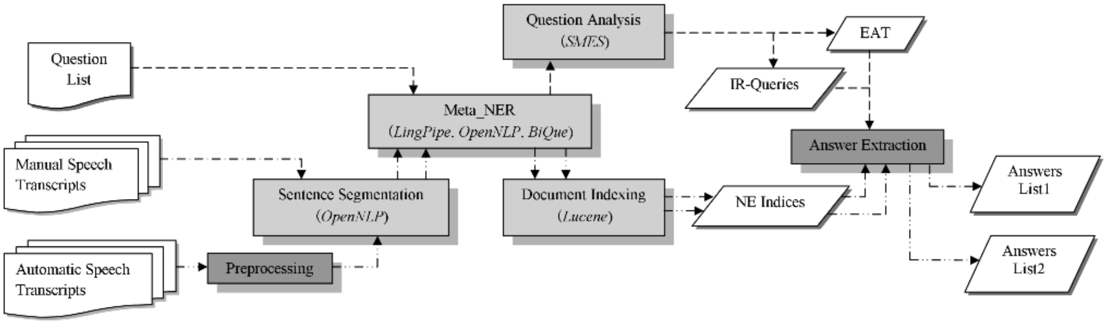

Figure 1: The architecture of QAst-v1.

The current architecture is shown in figure 1. In an off–line phase we firstly generate an inverted index for the speech corpora such that each sentence is considered as a single document and indexed by its word forms and named entities. In the question answering phase, a list of NL questions is passed to the system. Each NL question is analyzed by the named entity recognizer and by the question analysis component. The main output is a question object which represents the expected answer type (EAT) of the question and its relevant keywords. For example, the EAT of the question "Where is Southern Methodist University?" is LOCATION and the relevant keywords are "Southern Methodist University". From the question object an IR–query expression is created in order to access the indexed document space. The IR–query for the example question is {+neTypes:LOCATION AND +"southern methodist university"} which can be read as "select only documents (in our case only sentences) which contain at least one location entity and the phrase southern methodist university ", see section 5 for details. In the answer extraction step all found location names are considered as answer candidates and the most frequent answer candidates are selected as answers to the question, e.g., “Dallas” and “Texas” are found as possible answers in the manual transcript of the lecture corpus. For each question a list of its N–best answers is returned.

In the next sections, we describe the core components in more detail starting with the named entity recognition because it is used in all other components.

3 Named Entity Recognition

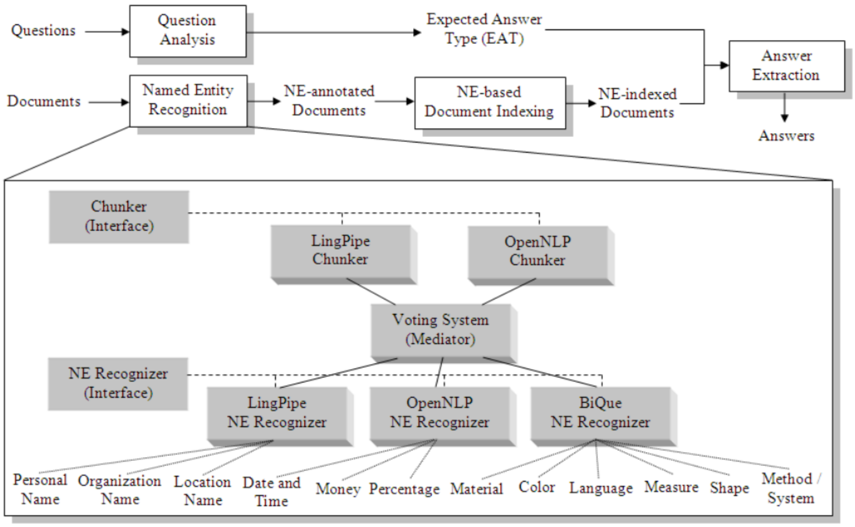

Figure 2: The internal structure or our hybrid named entity recognizer.

Named Entity Recognition (NER) plays a central role in a factual QA architecture: Named entities are the answers of factual questions and as such define the range for the expected answer types. In the QAst 2007 pilot task the following answer types are used: PERSON, LOCATION, ORGANIZATION, TIME, LANGUAGE, SYSTEM/METHOD, MEASURE, COLOUR, SHAPE, and MATERIAL. The answer types directly corresponds to the type of named entities which should be covered by NER.

There exists already a number NER components, but with different coverage of types. For that reason, we developed a hybrid NER approach where we combined three different NER components, see also figure 2:

- LingPipe developed by Alias–i and available at http://www.alias-i.com/lingpipe/. It mainly covers PERSON, LOCATION, and ORGANIZATION names for English and co–references between pronouns and corresponding named entities. It realizes a supervised statistical based approach to NER.

- Opennlp tools available at http://opennlp.sourceforge.net/. Its name finder is also based on a supervised statistical approach and covers mainly seven types of NEs for English, viz. PERSON, LOCATION, ORGANIZATION, DATE, TIME, MONEY, and PERCENTAGE.

- BiQueNER developed by our group. It is based on the co-training algorithm for named entities developed by [1]. We are using BiQueNER for handling all NE types, especially LANGUAGE, SYSTEM/METHOD, MEASURE, COLOUR, SHAPE, and MATERIAL.

LingPipe and Opennlp require large sets of annotated training material which we had not at hand for speech transcripts. Hence we directly used the language models both tools come with and which have been created especially for written news texts. BiQueNER embeds a very flexible gazetteer sub-component which we used for creating NE lists for the covered types using a simple word–based frequency analysis of the corpora and resources from Wikipedia. Performing a training step with BiQueNER turned out to be unfeasible for the moment, because BiQueNER requires that the corpus is preprocessed with a POS–tagger and a chunker. We used the Opennlp tools for this but the result of the preprocessing step on the CHILL corpora are currently inappropriate when using the available models of Opennlp. Linguistically annotated speech transcripts are needed urgently.

All three NERs run in parallel on an input text and we construct NE–specific indexes for each individual recognizer. The individual results are combined via the IR–query construction process (see sec. 5) and the answer extraction process (see sec. 6). In this way, also conflicting cases are handled like different NE readings (e.g., the same instance is typed as PERSON by LingPipe and ORGANIZATION by BiQueNER) and (implicit) partial or overlapping annotations (e.g., the name "Rui Wang" is recognized as a single PERSON name by LingPipe or as two different PERSON names by BiQueNER).

4 Document Preprocessing

Based on a number of experiments we made during the development of our textual open–domain QA–technology, we developed the hypothesis that a structural analysis of un-structured documents towards the information needs of questions, will support the retrieval of relevant small textual information units through informative IR-queries. However, since we cannot foresee all the different users' interests or questions, a challenging research question is: How detailed can the structural analysis be made without putting over a "straitjacket" of a particular interpretation on the unstructured source? Thus, there is a tradeoff between off-line and on-line document annotation. Questions and answers are somewhat related in that questions influence the information geometry and hence, the information view and access, cf. [4]. Furthermore, a sentence-oriented preprocessing determining only sentence boundaries, named entities (NE) and their co-references turned out to be a useful level of offline annotation of written texts, at least for the Clef–kind of factual questions.

For that reason we decided to apply the same off–line preprocessing approach also to the QAst collections. In particular the following steps are performed:

- Extracting lines of words from the automatic speech transcripts so that both the manual and automatic transcript are in the same format.
- Identification of sentence boundaries using the sentence splitter of the Opennlp tool which is based on maximum entropy modeling. We are currently using the language model the sentence splitter comes with which is optimized for written texts.
- Annotation of the sentences with recognized named entities.

The preprocessed documents are further processed by the IR–development engine Lucene, cf. [2]. We are using Lucene in such a way that for all extracted named entities and content words, Lucene provides indexes which point to the corresponding sentences directly. Especially in the case of named entities type–based indexes are created which support the specification of type constraints in an IR–query. By doing this, we could query the IR component not only by keywords extracted from the questions, but also by NE types corresponding to their expected answer types. An example would make this clear: for the question "Where is Southern Methodist University located?" beside creating an IR-query containing the keywords: {+"Southern Methodist University", +located}, we could supply also the expected answer type LOCATION querying an additional field neTypes: {+text:"Southern Methodist University", +text:located +neTypes:LOCATION}. This will not only narrow the amount of data being analyzed for answer extraction, but will also guarantee the existence of an answer candidate.

5 Question Processing and Sentence Retrieval

In the current QAst 2007 task setting natural language questions are specified in written form. For this reason we were able to integrate the question parser from our textual QA–system into QAst-v1. The question parser computes for each question a syntactic dependency tree (which also contains recognized named entities) and semantic information like question type, the expected answer type, and the question focus, cf. [3] for details. The semantic information is determined on the basis of syntactic constraints applied on relevant NP and VP phrases of the dependency tree, and by taking into account information from two small structured vocabularies. They basically perform a mapping from linguistic entities to values of the questions type, e.g., trigger phrases like *name of*, *type of*, *abbreviation of* or they perform a mapping of lexical elements to expected answer types, like *town*, *person*, *president*.

In a second step the result of the question parser is mapped to an ordered set of alternative IR–queries following the same approach as in our textual QA system, cf. [3]. The alternative IR–queries differ in the degree of their specialization. For example, for the question "Where is Southern Methodist University located?" we construct {+text:"Southern Methodist University", +text:located +neTypes:LOCATION} and {text:Southern text:Methodist text:University, text:located +neTypes:LOCATION}. The latter means that we relax the requirement that the phrase "Southern Methodist University" and the keyword "located" must both be present in a sentence to serve as a relevant snippet from which answer candidates are extracted.

The different IR–queries are passed to Lucene in the order of their specificity such that if a higher–ranked IR–query does not return any sentence, the next less specific IR–query is tried. This means that the alternative IR–queries are processed in the manner of a decision list. Lucene returns a list of all matching sentences. This means that each returned sentence at least contains one named entity which is an instance of the EAT and overlaps with the set of keywords depending on the specificity of the IR–query. The set of sentences are ordered according to Lucene'S similarity measure applied on each sentence and the IR–query

6 Answer Extraction

In our current version of QAst-v1 answers are considered as instances of the expected answer type (EAT). The EAT corresponds to a type of the named entities covered by our system. Thus we consider each named entity of the retrieved sentences as answer candidate if and only if its type is the same as the EAT. For each answer candidate we compute its frequency relative to the set of retrieved sentences and order them accordingly. We finally filter out ill–formed answer candidates if they consists of material like "the", "muh", "uhm" using a set of manually specified rules. If no EAT–compatible named entity exists the empty answer NIL is returned.

7 Results and Discussion

We took part in the tasks:

- T1: Question-Answering in manual transcriptions of lectures;
- T2: Question-Answering in automatic transcriptions of lectures;

In both cases the CHILL corpus was used which was adapted by the organizers for the QAst 2007 track. It consists of around 25 hours (around 1 hour per lecture) both manually and automatically transcribed. The language is European English, mostly spoken by non–native speakers.

We submitted only one run to each task and the table below shows the results we obtained:

Run	task	Questions returned (#) [98]	Correct answers (#)	MRR	Accuracy
dfki1_t1	T1	98	19	0.17	0.15
dfki1_t2	T2	98	9	0.09	0.09

where MRR is the Mean Reciprocal Rank that measures how well ranked is the right answer in the list of 5 possible answers in average. Accuracy is the fraction of correct answers ranked in the first position in the list of 5 possible answers.

The currently low number of returned correct answers has two main error sources. On the one hand side, the coverage and quality of the named entity recognizers are low. This is probably due to the fact that we used the languages models that were created from written texts. One possible solution is to improve the corpus preprocessing step, especially the sentence splitter and the repairment of errors like word repetition. Another possible source of improvement is the development of annotated training corpus of speech transcripts for named entities. Both activities surely demand further research and resources.

On the other hand side, the performance of the answer extraction process strongly depends on the coverage and quality of the question analysis tool. We will improve this by extending the current coverage of the English Wh–grammar, especially by extending the mapping of general verbs and nouns to corresponding expected answer types and by exploiting strategies that validate the semantic type consistency between the relevant nouns and verbs of a question.

Acknowledgement

The work presented here has been partially supported by a research grant from the German Federal Ministry of Education, Science, Research and Technology (BMBF) to the DFKI project HyLaP (FKZ: 01 IW F02) and by the EU funded project QALL-ME (FP6 IST-033860).

References

[1] M. Collins and Y. Singer. Unsupervised models for named entity classification, 1999.

[2] Erik Hatcher and Otis Gospodnetic. *Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA, 2004.

[3] G. Neumann and S. Sacaleanu. Experiments on robust nl question interpretation and multi-layered document annotation for a cross-language question/answering system. In *Clef 2004*, volume 3491, pages 411–422. Springer-Verlag LNCS, 2005.

[4] C.J. Van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Pres, 2004.

[5] B. Sacaleanu and G. Neumann. Dfki-lt at the clef 2006 multiple language question answering track. In *Working notes of Clef 2006*. August, Alicante, Spain.