Proceedings of the Australasian Language Technology Workshop 2007, pages 57-65

Named Entity Recognition in Question Answering of Speech Data

Diego Mollá **Menno van Zaanen** **Steve Cassidy**
Division of ICS
Department of Computing
Macquarie University
North Ryde
Australia
{diego,menno,cassidy}@ics.mq.edu.au

Abstract

Question answering on speech transcripts (QAst) is a pilot track of the CLEF competition. In this paper we present our contribution to QAst, which is centred on a study of Named Entity (NE) recognition on speech transcripts, and how it impacts on the accuracy of the final question answering system. We have ported AFNER, the NE recogniser of the AnswerFinder question-answering project, to the set of answer types expected in the QAst track. AFNER uses a combination of regular expressions, lists of names (gazetteers) and machine learning to find NeWS in the data. The machine learning component was trained on a development set of the AMI corpus. In the process we identified various problems with scalability of the system and the existence of errors of the extracted annotation, which lead to relatively poor performance in general. Performance was yet comparable with state of the art, and the system was second (out of three participants) in one of the QAst subtasks.

1 Introduction

AnswerFinder is a question answering system that focuses on shallow semantic representations of questions and text (Mollá and van Zaanen, 2006; van Zaanen et al., 2007). The underlying idea of AnswerFinder is that the use of semantic representations reduces the impact of paraphrases (different wordings of the same information). The system uses symbolic algorithms to find exact answers to questions in large document collections.

The design and implementation of the AnswerFinder system has been driven by requirements that the system should be easy to configure, extend, and, therefore, port to new domains. To measure the success of the implementation of AnswerFinder in these respects, we decided to participate in the CLEF 2007 pilot task of question answering on speech transcripts (QAst). The task in this competition is different from that for which AnswerFinder was originally designed and provides a good test of portability to new domains.

The current CLEF pilot track QAst presents an interesting and challenging new application of question answering. The objective in QAst is to answer questions based on transcripts of meetings and lectures. Both automatic and manual transcripts are provided; the automatic transcripts being the result of applying a speech recogniser to the audio recordings. The data for the task is taken from corpora collected by the AMI (Augmented Multiparty Interaction) project (Carletta et al., 2005) and from the CHIL (Computers in the Human Interaction Loop) project (Waibel et al., 2004). While both corpora are extensively annotated, only speaker turn annotation is provided in the input data for this task.

In our contribution we focus on adapting AFNER, our Named Entity Recogniser (NER), for speech transcripts and its application for Question Answering. Named Entity (NE) recognition is the task of finding instances of specific types of entities in free text. This module is typically one of the most impor-

tant sources of possible answers available to QA systems and therefore an improvement on its accuracy should result on an improvement of the accuracy of the complete QA system.

The AFNER system, like the AnswerFinder system, was designed with flexibility in mind. Since the properties of the NE recognition task in this competition are in several respects quite different to those of the task AFNER was originally designed for (as discussed in section 3.3), the QAst competition also allows us to measure the success of our AFNER implementation according to the configurability and extensibility criteria.

2 Question Answering on Speech Transcripts

The task of Text-Based Question Answering (QA) has been very active during the last decade, mostly thanks to the Question Answering track of the Text REtrieval Conference (TREC) (Voorhees, 1999). The kinds of questions being asked range from fact-based questions (also known as factoid questions) to questions whose answer is a list of facts, or definitions. The methods and techniques used have converged to a prototypical, pipeline-based architecture like the one we will describe here, and only recently the task has been diversified to more complex tasks such as TREC's QA task of complex interactive question answering (Dang and Lin, 2007) or the Document Understanding Conference (DUC)'s track of query-driven summarisation (Dang, 2006).

Whereas the TREC competitions concentrate on searching in English texts, CLEF (Cross-Language Evaluation Forum) focuses on non-English, cross-lingual and multi-lingual search. Within this forum several competitions are organised. The QAst track deals with question answering on speech data.

Prior to the QAst pilot track of CLEF there has been very little work on the area of question answering of speech data. Much of the work has focused on the task of recognising named entities by applying machine learning using features that leverage the very special kinds of information of speech data, particularly the lack of punctuation and capitalisation information. The work by Surdeanu et al. (2005) is an example of such an approach. Another line of work tries to recover the lost capitalisation information by using machine learning methods trained on regular text and tested on text where all capitalisation information has been removed. This is the approach followed, for example, by Li et al. (2003). Note, however, that Li et al. did not work on speech data as we are trying to do here but on regular text where case information has been removed. As we discuss below, speech data have many other factors that need to be taken into consideration.

Two data sets were provided by CLEF for development of systems participating in the evaluation. These were transcripts of lectures taken from the CHIL (Waibel et al., 2004) project and meetings from the AMI (Carletta et al., 2005) project. We made use of the AMI data because we had access to the original annotations which included named entities. This data consists of transcripts of 35 meetings each with up to four speakers. These contained around 254,000 words of dialogue. Due to disk space constraints we only made use of 15 meetings containing around 160,000 words in the development of our system.

2.1 AnswerFinder

The AnswerFinder question answering system is essentially a framework consisting of several phases that work in a sequential manner. For each of the phases, a specific algorithm has to be selected to create a particular instantiation of the framework. The aim of each of the phases is to reduce the amount of data the system has to handle from then on. This allows later phases to perform computationally more expensive operations on the remaining data.

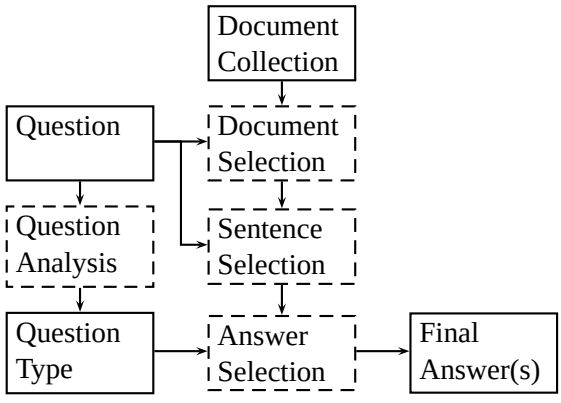

Figure 1: AnswerFinder system overview

Figure 1 provides an overview of the An-

swerFinder framework. The first phase is a document retrieval phase that selects documents relevant to the question. AnswerFinder was developed to work on large document collections and this phase can make a significant reduction in the amount of text that must be handled in subsequent steps.

Next is the sentence selection phase which selects a subset of sentences from the relevant documents selected in the previous phase. During sentence selection, all sentences that are still left (e.g. all sentences in the selected documents in the first step) are scored against the question using a relevance metric. The most relevant sentences according to this metric are kept for further processing. This phase can be applied to the remaining sentences several times using different metrics, each time reducing the number of sentences further.

After sentence selection, the remaining sentences are passed to the answer selection phase. The answer selection phase aims at selecting the best of the possible answers to return to the user. In the experiments described here, the list of possible answers is generated by applying a NER to the remaining sentences.[1]

Next, the question is analysed, providing information about the kind of answer that is required. From the possible answers, those that match the type of answer required by the question are selected and scored. Finally, the best answer is returned to the user. Best answer in this context is considered to be the answer that has both the highest score and an answer type that matches the question, or simply the answer with the highest score if none of the possible answers fit the expected answer type.

2.2 Applying AnswerFinder to Speech Transcripts

Question answering on speech transcripts introduces specific challenges compared to text-based QA due to the nature of the genre and the process of transcription. AnswerFinder has been initially developed to work on news articles which are typically well-written pieces of text. The casual, multi-party spoken language used in this evaluation is very different. For example,

- There are frequent false starts and sentences that are interrupted in the discourse.
- There are filling words that usually do not appear in free text (and in particular news text), such as “er”, “uh”, etc. In our experiments, this is particularly problematic when these words appear inside a named entity, e.g. “Rufford, um, Sanatorium, that’s right”.
- The grammatical structure of the transcription does not conform to that of free text. Consequently most tools, such as parsers and chunkers, which would normally be used in specific AnswerFinder phases, produce very poor results.
- If the transcript is an automatic transcript (produced by a speech recogniser) there are errors of transcription and missing information, most notably punctuation characters and capitalised characters. This information is used in many phases of AnswerFinder when answering questions on news data.
- During training, a corpus annotated with named entities is used. The density of named entities in free speech is much smaller than in usual corpora (containing news text).

Many of the above features make it difficult to do traditional linguistic processing such as parsing and semantic interpretation. For this reason, many of the instantiations of the phases we have implemented, which typically use complex linguistic processing (as described in van Zaanen et al. (2007)) would not perform well. We consequently decided not to use some of AnswerFinder’s more linguistically-intensive modules. Instead we focused on attempting to increase the accuracy of the task of recognition of named entities. Thus, the question answering method used for QAst is entirely based on the task of finding and selecting the right entities.

In particular, the instantiation of the AnswerFinder framework that generated the QAst 2007 results consists of the following algorithms for the phases:

[1] In general, some sentence selection methods have the ability to generate possible answers that can also be selected during the answer selection phase. However, these algorithms are not used in these experiments as will be discussed in section 2.2.

- The document selection component returns the full list of documents provided for the complete list of questions. The total number of documents provided by the organisers of QAst is fairly small and therefore the other components of AnswerFinder are able to handle all documents. Essentially no documents are preselected in this instantiation. We do not attempt to rank the documents in any way.

- As a pre-processing step, the named entity recogniser is run over all the documents. This allows for more efficient handling of the set of questions, as named entity recognition only has to occur once.

- The sentence selection component is based on the word overlap between the question and the document sentences. This metric counts the number of words that can be found in both question and sentence after removing stop words. A simple sentence splitter method is used, which relies on the existence of punctuation marks when available, or on speech turns. Only sentences that contain NEs of the required type are considered.

- Each of the named entities found in the selected sentences are scored. The score of a NE is the sum of the number of occurrences of that NE with a particular type.

- The question classification component is based on a decision list of hand-constructed patterns of regular expressions. Each regular expression determines a question type and consequently a set of NE types.

- The answer extraction component selects five NEs that are of the expected answer type and have the highest NE scores. QAst allows for the system to return up to five answers. If four or fewer NEs of the correct type are found, then a NIL answer (meaning no answer) is returned as an option after presenting all found NEs. If no NEs of the expected type are found at all, the returned answer is NIL.

3 AFNER

Within the AnswerFinder project, we recently incorporated a purpose-built NER, called AFNER (Mollá et al., 2006). This NER has been specifically designed for the task of QA. AFNER differs from other NERs in that it aims to increase recall of recognition of entities, at the expense of a possible loss of precision (Mollá et al., 2006; van Zaanen and Mollá, 2007). Crucially, it allows the allocation of multiple tags to the same string, thus handling the case of ambiguous entities or difficult entities by not committing to a single tag. The rationale is that we do not want to remove the right answer at this stage. Instead we let the final answer extraction and scoring mechanism make the final decision about what is a good answer.

AFNER is ultimately based on machine learning. We use a maximum entropy classifier, and the implementation of this classifier is adapted from Franz Josef Och's *YASMET*[2]. Obviously, the selection of the features used in the classifier is very important.

3.1 Features

The features used by AFNER combine three kinds of information: regular expressions, gazetteers, and properties internal and external to the token. These features are described in more detail elsewhere (Mollá et al., 2006; van Zaanen and Mollá, 2007) and we will only briefly present them here.

The regular expressions used in AFNER are manually created and are useful for identifying strings that match patterns that are characteristic to entity types such as dates, times, percentages, and monetary expressions. These types of named entities are relatively standardised and therefore easy to find with high precision. However, the range of entities that can be discovered using regular expressions is limited. Matching a particular regular expression is a key feature used in identifying entities of these particular types.

Gazetteers are useful for finding commonly referenced entities such as names. AFNER uses three lists (locations, person names, and organisations), with a total of about 55,000 entries. The occurrence of tokens in one of the gazetteers is incorporated in the machine learning component. This allows for,

[2] `http://www.fjoch.com/YASMET.html`

Regular Expressions	Specific patterns for dates, times, etc
FoundInList	The token is a member of a gazetteer
InitCaps	The first letter is a capital letter
AllCaps	The entire word is capitalised
MixedCaps	The word contains upper case and lower case letters
IsSentEnd	The token is an end of sentence character
InitCapPeriod	Starts with capital letter and ends with period
OneCap	The word is a single capitalised letter
ContainDigit	The word contains a digit
NumberString	The word is a number word ('one', 'thousand', etc.)
PrepPreceded	The word is preceded by a preposition (in a window of 4 tokens)
PrevClass	The class assigned to the previous token
ProbClass	The probability assigned to a particular class in the previous token
AlwaysCapped	The token is capitalised every time it appears

Table 1: A selection of features used in AFNER

for example, context information in the final decision of the tag assignment for that particular token.

Finally, there are three types of features that relate to specific aspects of the separate tokens. The first type focuses on internal evidence and highlights token properties including capitalisation, alpha/numeric information, etc. Some specific features are listed in Table 1.

The second type of features focuses on external evidence that relates a token to tokens in surrounding text. Features that indicate which class has been assigned to the previous tokens and all of its class probabilities are also part of this type of feature.

The last type of features focuses on global evidence related to all occurrences of the same token. These features are mainly inspired on features described by Chieu and Ng (2002). Currently AFNER only checks whether a token is always capitalised in a passage of text.

3.2 General Method

The features described in the previous section are used in a maximum entropy classifier which for each token and for each category computes the probability of the token belonging to the category. Categories in this case are the named entity types prepended with 'B' and 'I' (indicating whether the token is at the beginning or inside a NE respectively), and a general 'OUT' category for tokens not in any entity. So for n named entities, $n * 2 + 1$ categories are used.

The classifier returns a list of tags for each token ordered based on probability. We select only those tags that have a probability of more than half of the probability of the next tag in the list. This initial threshold already removes tags that have a low probability. However, we also only allow a certain maximum number of tags to pass through. Preliminary experiments revealed that often the top two or three tag probabilities have similar values, but that tags lower down the list still pass the initial threshold, while they are not correct. By setting a threshold that limits the maximum number of tags to be returned we also filter those out. The results presented in this paper are generated by setting the second threshold to allow two tags per token. Initial experiments showed that this increases recall considerably. Allowing more tags increases recall only slightly while decreasing precision considerably.

Once tokens are assigned tags, they are combined to produce the final list of multi-word NEs as described elsewhere (Mollá et al., 2006; van Zaanen and Mollá, 2007). The result is an assignment of named entities to the sequence of tags where the named entities may be nested. This way we aim at high recall by allowing multiple interpretations of problematic strings that could be ambiguous.

Class	Type	# in BBN	# in AMI
ENAMEX	Language	9	0
	Location	2,468	16
	Organization	4,421	27
	Person	2,149	196
	System	0	448
	Color	0	283
	Shape	0	147
	Material	0	267
TIMEX	Date	3,006	9
	Time	96	147
NUMEX	Measure	2,568	293
	Cardinal	0	646

Table 2: Named Entities used for QAst. The numbers of entities listed in the two last columns refer to the actual training set (a subset of BBN and AMI).

Class	Type
ENAMEX	Organization
	Person
	Location
TIMEX	Date
	Time
NUMEX	Money
	Percent

Table 3: Entity types used in the original version of AFNER

3.3 Adaptation of AFNER to QAst

AFNER has been developed to work on news data, and as such, we had to modify parts of the system to allow it to be used in the QAst task. The first adaptation of AFNER is the selection of NE types. Originally AFNER focused on a limited set of entities similar to those defined in the Message Understanding Conferences (Sundheim, 1995), and listed in Table 3.

For QAst we used a set of entity types that closely resembles the kinds of answers expected, as described by the QAst 2007 specification. The types used by the modified AFNER are listed in Table 2.

The regular expressions that are used in AFNER to find MUC-type named entities were extended to cover the new types of entities. This process did not require much additional work, other than adding a few common names of shapes and colours. The lists of names that was part of the initial AFNER was left untouched.

The general machine learning mechanism was left unmodified, and the set of features was also left untouched. The only difference was the choice of training corpus. We mapped the annotated entities of the BBN corpus that we had used previously, and added a fragment of the development set of the AMI corpus.

However, due to problems of scalability during training (the intermediate files produced were very large due to the increased number of classifier categories) we were not able to use all the files. For these experiments we used 26 documents from the AMI corpus and 16 from the BBN corpus. Table 2 shows the total number of entities annotated in the BBN and the AMI parts of the training set. The entity types of each kind of corpus complement each other, though some of the entity types had few instances in the corpora, most notably, the type Language only occurred nine times.

We decided to use the BBN corpus to complement the annotations of AMI because some entity types that were very scarce in AMI were very common in BBN. Also, the entity types annotated in AMI are not the sort of types that would typically be annotated as named entities. For example, the entity type “Person” would have instances like *industrial designer*. Furthermore, the quality of some of the annotations of the AMI corpus was poor. In at least

two of the 26 meetings the contents of named entities seemed to be random strings. After submitting the results, we found a bug in our corpus processing script which resulted in some named entities having extra words included in them.

4 Results

We participated in all the QAst tasks, which are described below:

CHIL$_M$ Manual transcripts from the CHIL corpus of lectures;

CHIL$_A$ Automated transcripts from the CHIL corpus;

AMI$_M$ Manual transcripts from the AMI corpus of meetings; and

AMI$_A$ Automated transcripts from the AMI corpus.

We provided two runs per task. We were interested on determine the impact of the machine learning component of AFNER. Given the reduced number of training documents and the existence of errors in some of them we expected that the machine learning component would not be useful. Thus, the first run used the full AFNER system, whereas the second run (named "noML") used a version of AFNER that had the machine learning component disabled (essentially only using the regular expressions and the gazetteers). The results are shown in Table 4.

The results returned by CLEF indicate, as expected, comparatively poor performance with respect to the other participants. We are pleased to notice, however, that the results of task CHIL$_M$ are second best (from a group of three participants). Task CHIL$_M$ is the task that used the AMI transcripts and it was the task that we used to develop and fine-tune the system. The other tasks simply used the same settings. We are particularly pleased to learn that the results of task CHIL$_M$ are higher than the results we obtained during development time. This is possibly due to the nature of our tuning experiments, since we automatically applied the answer patterns to the answers found, and it could have been the case that correct answers which happened not to match the patterns were automatically marked as incorrect in our experiments. The evaluations carried by CLEF used human judges so they would be able to detect correct answers that had an unusual format.

The results indicate that none of the differences in results between the full and the noML runs are statistically significant under the paired t-test. This confirms our suspicion that the machine learning component of AFNER was not helping the question answering process at all. The likely reason for this is, as described above, the small size of the training data and the existence of noise in the NE annotations of the AMI corpus.

Our method to handle NIL questions is simple yet relatively effective to the point that correct NIL answers were an important part of the correct answers. Task AMI$_A$ in particular, which has 15 NIL questions, results in a halved MRR (from 14.10% down to 7.05% in our noML run) when all NIL questions are removed. It is encouraging to observe that, even after removing all NIL questions, task CHIL$_M$ has relatively good results (from 26.39% down to 22.38% in our noML run). The results of the non-NIL questions are shown in Table 5.

5 Conclusions and Further Work

In our contribution to the QAst competition we reused as much as we could of AnswerFinder, our question answering system, and AFNER, our Named Entity recogniser. Due to the nature of the speech corpus we needed to simplify the processing done by AnswerFinder and made it rely more heavily on the entities found by AFNER. The whole experiment showed successfully that both AnswerFinder and AFNER are flexible and can be adapted easily to new tasks.

The small training corpus and the presence of annotation errors in the AMI corpus made the machine learning component of AFNER ineffective. An immediate line of further research is to investigate the cause of the errors, and correct them. Other lines of research are:

- Revise the machine learning component of AFNER, possibly replace it with another more scalable method, so that larger training corpora can be used. Currently we are investigating more efficient ways of storing the intermediate data.

Run	Questions	Correct Answers	MRR	Accuracy
full-CHIL$_M$	98	17.35%	9.98%	6.12%
noML-CHIL$_M$	98	16.33%	9.44%	5.10%
full-CHIL$_A$	98	14.29%	7.16%	3.06%
noML-CHIL$_A$	98	12.24%	5.88%	2.04%
full-AMI$_M$	96	35.42%	24.51%	16.67%
noML-AMI$_M$	96	33.33%	26.39%	20.83%
full-AMI$_A$	93	19.35%	11.24%	6.45%
noML-AMI$_A$	93	22.58%	14.10%	8.60%

Table 4: Results of the CLEF runs

Run	Questions	Correct Answers	MRR	Accuracy
full-CHIL$_M$	88	12.50%	8.56%	6.82%
noML-CHIL$_M$	88	11.36%	7.95%	5.68%
full-CHIL$_A$	87	5.75%	4.06%	3.45%
noML-CHIL$_A$	87	3.45%	2.87%	2.30%
full-AMI$_M$	86	29.07%	22.33%	18.60%
noML-AMI$_M$	86	25.58%	22.38%	19.77%
full-AMI$_A$	79	6.33%	3.90%	2.53%
noML-AMI$_A$	78	8.97%	7.05%	5.13%

Table 5: Results of non-NIL questions

- Review the features used for identifying the entities. Most of the current features rely on information about capitalisation, presence of digits, or punctuation marks but none of those are available on speech transcripts. In practice, using features that always provide the same values means that the machine learning component does not add much to the non-machine learning information, as shown in the experiment. More useful features will increase the use of the machine learning component.

- Use additional corpora. There are a few corpora of speech transcriptions available with annotations of named entities that we could use. Among the options is the corpus of speech transcripts within the SQUAD project with the UK Data Archive at the University of Edinburgh.

To conclude, question answering on speech transcripts is a challenging task that deserves greater attention by the research community. The CLEF QAst track is a step toward facilitating research on this area. Our participation in QAst is a step from our side to contribute to this exciting research area.

References

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain A. McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2005. The ami meetings corpus. In L. P. J. J. Noldus, F. Grieco, L. W. S. Loijens, and Patrick H. Zimmerman, editors, *Proceedings of the Measuring Behavior 2005 symposium on "Annotating and measuring Meeting Behavior"*. AMI-108.

Haoi Leong Chieu and Hwee Tou Ng. 2002. Named entity recognition: A maximum entropy approach using global information. In *Proceedings COLING 2002*.

Hoa Dang and Jimmy Lin. 2007. Different structures for evaluating answers to complex questions: Pyramids won't topple, and neither will human assessors. In *Proceedings ACL*.

Hoa Tran Dang. 2006. Duc 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Sydney. Association for Computational Linguistics.

Wei Li, Rohini Srihari, Cheng Niu, and Xiaoge Li. 2003. Question answering on a case insensitive corpus. In

Proc. ACL 2003 Workshop on Multilingual Summarization and Question Answering, pages 84–93.

Diego Mollá and Menno van Zaanen. 2006. Answerfinder at TREC 2005. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proc. TREC 2005*. NIST.

Diego Mollá, Menno van Zaanen, and Luiz A.S. Pizzato. 2006. Named entity recognition for question answering. In *Proceedings ALTW 2006*, page 8 pages.

Beth M. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Proc. Sixth Message Understanding Conference MUC-6*. Morgan Kaufmann Publishers, Inc.

Mihai Surdeanu, Jordi Turmo, and Eli Comelles. 2005. Named entity recognition from spontaneous open-domain speech. In *Proceedings Interspeech-05*, Lisbon.

Menno van Zaanen and Diego Mollá. 2007. A named entity recogniser for question answering. In *Proceedings PACLING 2007*.

Menno van Zaanen, Diego Mollá, and Luiz Pizzato. 2007. Answerfinder at trec 2006. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings TREC 2006*, page 8 pages.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In Ellen M. Voorhees and Donna K. Harman, editors, *Proc. TREC-8*, number 500-246 in NIST Special Publication. NIST.

A. Waibel, H. Steusloff, and R. Stiefelhagen. 2004. Chil - computers in the human interaction loop. In *5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisbon, Portugal.