AnswerFinder at QAst 2007: Named Entity Recognition for QA on Speech Transcripts

Diego Mollá, Steve Cassidy, Menno van Zaanen
Macquarie University
{diego,steve,menno}@ics.mq.edu.au

Abstract

Macquarie University's contribution to the QAst track of CLEF is centered on a study of Named Entity (NE) recognition on speech transcripts, and how such NE recognition 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 types of answer types expected in the QAst track. AFNER uses a combination of regular expressions, lists of names (gazetteers) and machine learning. The machine learning component is a Maximum Entropy classifier and was trained on a development set of the AMI corpus. Problems with scalability of the system and errors of the extracted annotation lead to relatively poor performance in general, though the system was second (out of three participants) in one of the QAst subtasks.

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; I.2.7 [**Natural Language Processing**]: Text Analysis

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Speech transcripts, Named Entity Recognition

1 Introduction

AnswerFinder is a question answering system that is being developed focusing on shallow semantic representations of questions and text [4, 5]. The underlying idea is that these semantic representations reduce the impact of paraphrases (different wordings of the same information). Overall, 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 QAst competition. The task in this competition is different from that for which AnswerFinder was originally designed. Applying the system to a new task would illustrate potential problems with respect to configurability and extensibility.

In addition, in our contribution we focused on the localisation of 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 important 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, just like the AnswerFinder system, was designed with flexibility in mind. Since the properties of the NE recognition task in this competition are quite different in several respects to that of which AFNER was originally designed, the QAst competition also allowed us to measure the success of our AFNER implementation according to the configurability and extensibility criteria.

2 Question Answering for Speech Transcripts

The task of Text-Based Question Answering (TBQA) has been very active during the last decade, mostly thanks to the Question Answering track of the Text REtrieval Conference (TREC) [9]. 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 [2] or the Document Understanding Conference (DUC)'s track of query-driven summarisation [3].

The present CLEF pilot track QAst presents an interesting and challenging new application of question answering, and in this contribution we have attempted to re-use as much as we could of AnswerFinder, a TBQA system that is designed for configurability, flexibility and portability to other domains. Part of our interest in participating in QAst was to test AnswerFinder's portability.

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.

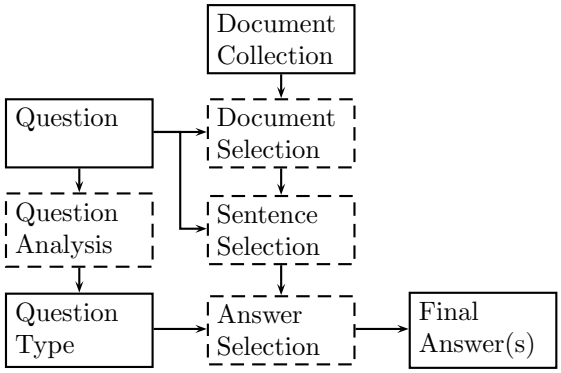

Figure 1: AnswerFinder system overview

Figure 1 provides an overview of the AnswerFinder framework. The first phase is a document retrieval phase that selects relevant documents. AnswerFinder was developed to work on large document collections and this phase typically reduces a great amount of text that will be handled in subsequent steps.

Next is the sentence selection phase. This can actually be a sequence of steps, each of 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.

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 provided by a NER.[1] Thus, 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 fits the expected answer type.

2.2 Applying AnswerFinder to Speech Transcripts

Question answering on speech transcripts introduces specific challenges compared to TBQA due to the nature of the genre and the process of transcription. AnswerFinder has been initially developed to work on news articles. News articles are typically well-written pieces of text. Analysing the documents in the QAst competition, it is clear that speech transcripts are 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 with 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 to answer questions on news data.
- When using a corpus annotated with named entities, the density of named entities in free speech is much smaller than in usual corpora.

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 (which are described in [5]) would not perform well. We consequently decided not to use AnswerFinder's syntactic and graph-semantic information. 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 AnswerFinder framework that generated the QAst 2007 results consists of the following instantiations of the phases:

- The document selection component returns the full list of documents provided for the complete list of questions. The total number of documents is fairly small and therefore the other components of AF are able to handle all documents. We do not attempt to rank the documents in any way.

[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 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.
- The question classification component is based on a decision list of hand-constructed patterns of regular expressions.
- The answer extraction component selects five NEs that are of the expected answer type and have the highest scores. If four or less NEs are found, then a NIL 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. The score of a NE is the sum of the individual scores of each occurrence of a NE. The individual score of a NE is the confidence of AFNER to label the answer candidate with the particular NE label.

3 AFNER

Within the AnswerFinder project, we recently incorporated a purpose-built NER, called AFNER [6]. This NER has been specifically designed for the task of TBQA. 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 [6, 8]. 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 weed out the right answer at this stage. Instead we let the final answer extraction 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.

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. If one or more words are found in one of the gazetteers, which are lists of names, locations, etc., then it is likely that the expression is of the type indicated by the list. However, this this is not always the case. For example, common person names may also be regular words (*Smith*, *Baker*). We use gazetteers as additional features in the classifier to increase the likelihood of these kinds of named entities. It also allows the classifier to use other features that combined may be more determinant for the categorisation of a specific token in particular entities. AFNER uses three lists (locations, person names, and organisations), with a total of about 55,000 entries.

Finally, there are three types of features that relate to specific aspects of the separate tokens. Firstly, we identify features that illustrate internal token properties including capitalisation, alpha/numeric information, etc. Some specific features are listed in Table 1.

Secondly, AFNER incorporates some contextual features. These are features that concentrate on the token in the surrounding text, or relate a token to tokens in surrounding text. These features

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

are implemented through a set of regular expressions that are matched against neighbouring tokens within a context window of the token under consideration. When a regular expression matches the context, this is recorded. These regular expressions detect patterns such as whether the neighbouring token is made of two digits, or whether it is a currency name. Features that consider the class assigned to the previous tokens and all of its class probabilities are also part of this type of feature.

Thirdly, there is a set of features that measure global information of the tokens. These features are mainly inspired on features described by [1]. Currently AFNER only checks whether a token is always capitalised in a passage of text.

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: Features used in AFNER

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 order. 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.

AFNER assigns multiple tags to tokens. By doing this, we aim for high recall. The presence of multiple tags also means that NEs can be nested, meaning that NEs may exist within other NEs.

Once tokens are assigned tags, they are combined to produce the final list of NEs. Each token that has a 'B' tag assigned to it is considered the beginning of a new NE of that type. All 'I' tags continue a NE if the previous token already had either a 'B' or 'I' tag of the same type assigned to it. If there was no such tag assigned to the previous token, the 'I' tag is taken to be a 'B' tag and indicates the start of a new NE of that type. Additionally, if a 'B' tag is preceded by a token with an 'I' tag, it will be taken to be both as a separate entity (with the previous entity ending) and as a continuation of the previous token.

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

Table 2: Entity types used in the original version of AFNER

The result of this algorithm is an assignment of named entities to the sequence of tags where the named entities may overlap, as is illustrated in Figure 2.

Figure 2: Named entities as multiple labels. The token-based labels appear above the words. The final NE labels appear below the words.

To compute the probability of a sequence of tokens (with corresponding named entity types), we use the geometric mean. This is done to normalise over the length of the named entities. The computation works as follows. Take P_i to be the probability of token i and $P_{1...n}$ the probability of the entire list of tokens (from begin to end). The geometric mean of the probabilities is computed as:

$$P_{1...n} = e^{\frac{\sum_{i=1}^{n} \log P_i}{n}}$$

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 [7], and listed in Table 2.

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 3.

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 classes the classifier can return) we were not able to

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 3: 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).

Run	Questions	Correct Answers	MRR	Accuracy
clt1-t1	98	17.35%	9.98%	6.12%
clt2-t1	98	16.33%	9.44%	5.10%
clt1-t2	98	14.29%	7.16%	3.06%
clt2-t2	98	12.24%	5.88%	2.04%
clt1-t3	96	35.42%	24.51%	16.67%
clt2-t3	96	33.33%	26.39%	20.83%
clt1-t4	93	19.35%	11.24%	6.45%
clt2-t4	93	22.58%	14.10%	8.60%

Table 4: Results of the CLEF runs

use all the files. For these experiments we used 26 documents from the AMI corpus and 16 from the BBN corpus. Table 3 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 were very scarce in AMI but 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 very bad, to the point that, for example, the entity type "Color" would have instances like *fancy* or *uh*, and even just punctuation marks such as commas or periods. The later errors of annotation make us suspect that perhaps the process to extract the entities from the AMI corpus, which was very laborious, had one mistake or two. We plan to revise the full process of extraction and re-do the experiments.

4 Results

We participated in all the QAst tasks and provided two runs per task. The first run used the full AFNER system, whereas the second run used a version of AFNER that had the machine learning component disabled. 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 3 are second best (from a group of three participants). Task 3 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 1, 2, and 4

Run	Questions	Correct Answers	MRR	Accuracy
clt1-t1	88	12.50%	8.56%	6.82%
clt2-t1	88	11.36%	7.95%	5.68%
clt1-t2	87	5.75%	4.06%	3.45%
clt2-t2	87	3.45%	2.87%	2.30%
clt1-t3	86	29.07%	22.33%	18.60%
clt2-t3	86	25.58%	22.38%	19.77%
clt1-t4	79	6.33%	3.90%	2.53%
clt2-t4	78	8.97%	7.05%	5.13%

Table 5: Results of non-NIL questions

simply used the same settings. We are particularly pleased to learn that the results of task 3 are higher than the results we obtained during development time. This is possibly due to the nature of our 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.

Our preliminary experiments indicated that the machine learning component was not helping the question answering process at all. The CLEF results show some increase of correct answers in the first run (with machine learning) in the tasks based on the CHIL corpus (tasks 1 and 2) but a decrease of correct answers in the tasks based on the AMI corpus (tasks 3 and 4). Our preliminary experiments used the AMI corpus only, and therefore the results are consistent with our experiments. Given the poor overall results with the CHIL corpus it is reasonable to suspect that the patterns and lists do not do well with the CHIL corpus and therefore the machine learning component can help. The patterns and lists were not fine-tuned either for CHIL or for AMI, they were simply the ones we used for the original, news-based BBN text corpus. We will investigate the relative impact of the patterns and lists on one side and the machine learning component on the other side for the speech transcripts.

Our method to handle NIL questions is simple yet relatively effective to the point that correct NIL answers were a significant part of the correct answers. Task 4 in particular, which has 15 NIL questions, results in a halved MRR (from 14.10% down to 7.05% in our second run) when all NIL questions are removed. Still, task 3 has relatively good results after removing all NIL questions (from 26.39% down to 22.38% in our second run). The results of the non-NIL questions are shown in Table 5.

5 Conclusions and Further Work

In our contribution to QAst 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 simplified the processing done by AnswerFinder and made it rely more heavily on the entities found by AFNER. The whole experiment showed 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.
- 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.

- 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.

References

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

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

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

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

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

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

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

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

[9] Ellen M. Voorhees. 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, 1999.