The University of Sheffield's TREC 2006 Q&A Experiments

Mark A. Greenwood, Mark Stevenson and Robert Gaizauskas
{m.greenwood, m.stevenson, r.gaizauskas}@dcs.shef.ac.uk

Department of Computer Science
University of Sheffield, UK

1 Introduction

As a natural language processing group (NLP) our original approach to question answering was linguistically motivated culminating in the development of the QA-LaSIE system (Humphreys et al., 1999). In its original form QA-LaSIE would only propose answers which were linked via syntactic/semantic relations to the information missing from the question (for example *"Who released the Internet worm?"* is missing a person). While the answers proposed by the system were often correct, the system was frequently unable to suggest any answer. The next version of the system loosened the requirement for a link between question and answer which improved performance (Scott and Gaizauskas, 2000). There are still a number of open questions from the development of the QA-LaSIE system: does the use of parsing and discourse interpretation to determine links between questions and proposed answers result in better performance than simpler systems which adopt a shallower approach? Is it simply that the performance of our parser is below the level at which it could contribute to question answering? Are there questions which can only be answered using deep linguistic techniques? With the continued development of a second QA system at Sheffield which uses shallower techniques (Gaizauskas et al., 2005) we believe that we are now in a position to investigate these and related questions. Our entries to the 2006 TREC QA evaluation are designed to help us answer some of these questions and to investigate further the possible benefits of linguistic processing over shallower techniques.

The remainder of this paper is organised as follows. Firstly the framework in which our systems are developed is described in section 2 along with the QA system components. Section 3 describes the configurations and aims of our evaluation runs. Section 4 discusses the official evaluation results of our submitted runs in relation to the research questions outlined above.

2 QA Framework

To simplify both the development and evaluation of our multiple approaches to question answering the separate components are hosted within a newly developed framework. This framework abstracts away from specific implementations of the components within a QA system. This allows us to develop multiple competing components which can easily be substituted for one another within a QA system. The framework also allows us to provide a number of common processing resources which not only simplifies the development of the more complex components, but also ensures the consistent lower level treatment of texts to allow for valid performance comparisons between competing components.

The remainder of this section details both the common processing available as well as the component implementations used by the runs submitted for evaluation which are described in Section 3.

2.1 Common Text Processing

There are a number of language processing tools that most (if not all) question answering systems will require. These include relatively low level tasks, such as tokenization and sentence splitting, as well as more complex task, such as named entity recognition (or as we refer to the more abstract case - semantic entity tagging). The framework provides these resources by utilizing the GATE framework (Cunningham et al., 2002), also developed within the NLP group at Sheffield. Specifically we utilize extended versions of the ANNIE components to allow tokenization, sentence splitting, part-of-speech tagging, morphological analysis, gazetteer lookup and semantic entity recognition. This common set of processing components ensures that the same basic information can be extracted from the examined texts irrespective of the approach to question answering used. This makes comparisons between differing QA approaches easier to carry out and their results more reliable.

Whilst these components provide a solid foundation for our QA systems it does not address the problem of there being multiple ways of representing identical pieces of information. The answers to many questions can be represented in many ways and as most QA systems rely, at least in part, on the frequency of occurrence of competing candidate answers (see e.g. Light et al. (2001) the ability to accurately compare candidate answers is important. To this end all dates and numbers are normalised to

a standard format. Dates are converted to a standard numerical format including resolving partial or descriptive dates (such as *today* or *tomorrow*) against the date of the newswire article. Numbers, both isolated and within measurements, are converted to a plain numeric form, e.g. *3000*, *3,000*, and *three thousand* are all represented as *3000*.

While such normalisation helps with the comparison of answers, and ultimately, in the ranking used to determine the most likely answer, it does not address issues of ambiguity in proposed answers. For example dates provided as answers can often be incomplete and ambiguous simply because of the way they appear in text. A newswire article discussing a recent or ongoing event will simply give the date as the day and month without the year as this is implicitly specified by the date of the document. We solve this specific issue by ensuring that when a date is proposed as an answer it is expanded to a full non-ambiguous form (i.e. contains a day, month and year) from the information present in the supported document.

2.2 Passage Retrieval with Lucene

All three of our runs use the open-source Lucene[1] IR engine to index and access the AQUAINT collection. Each document is split into separate paragraphs using the embedded SGML paragraph tags. All remaining SGML tags are then removed and each paragraph, after having been stemmed (Porter, 1980) and stopwords removed, is added to the Lucene index along with the unique document ID and associated date.

Our current approaches to answering factoid and list questions use the same approach to retrieve relevant documents for further processing. The question and target are first combined to form a single IR query (using the approach documented by Gaizauskas et al. (2005)) and then this query is used to retrieve the twenty most relevant passages from the AQUAINT collection. The use of only the top twenty passages is based on a number of experiments (Gaizauskas et al., 2003; Greenwood, 2006) carried out using combinations of various IR engines and QA systems which all suggest that while retrieving more text means greater coverage (i.e. the percentage of questions for which at least one relevant passage is retrieved (Roberts and Gaizauskas, 2004)) there comes a point at which the larger volumes of text actually inhibit the ability of answer extraction components to extract correct answers. Whilst this approach maximises our answer extraction performance (and hence the end-to-end performance of our QA systems) the coverage of the retrieved passages is approximately 52% (Greenwood, 2006) which means that the maximum accuracy the QA systems could achieve would also be 52% (i.e. we cannot answer a question if we do not retrieve any answer bearing documents).

2.3 Answering Factoid and List Questions

Once relevant documents have been retrieved we use two main approaches to answering factoid questions: a linguistic approach called QA-LaSIE and a shallow semantic tagging approach. Both can also be used to answer list questions simply by returning more than one answer[2].

2.3.1 QA-LaSIE

QA-LaSIE has traditionally been our main approach to question answering and it has been used in each TREC QA evaluation in which we have participated. QA-LaSIE performs partial syntactic and semantic analysis of questions and candidate answer bearing documents and then performs matching over a derived logical form representation. The system has been described in detail in past TREC proceedings (Greenwood et al., 2002) and does not differ substantially to the version used in the 2005 TREC evaluation (Gaizauskas et al., 2005).

2.3.2 Semantic Tagger

For TREC 2003 (Gaizauskas et al., 2003) we introduced a simple baseline system for answering factoid and list questions based around the premise that an answer to a question will be an entity from a fixed set of semantic types. Since its introduction this system has improved to the point where it is now consistently outperforming QA-LaSIE and is no longer considered a baseline system.

This system consists of two main components: a rule based question analyser and a semantic tagger based answer extraction component. The development of this system is documented in some detail by Greenwood (2006) for a brief overview see Gaizauskas et al. (2005).

When using this approach to answer list questions we were previously heavily penalised for not returning any answers to questions for which the semantic type of the expected answer could not be accurately determined. In the 2005 evaluation (Gaizauskas et al., 2005) we experimented with a very simple system which guessed answers based on frequency of occurrence of base noun phrases. This system has also been incorporated into the framework and is again used to answer list questions when the semantic tagging approach fails to find any answers.

[1] `http://lucene.apache.org/`

[2] If less than ten answers are found then all are returned as answers to the list question otherwise the first ten answers are returned along with any others which have a score above 0.08 (chosen by empirical testing over questions from previous TREC evaluations).

Run Tag	Factoid	List	Other	Combined
shef06qal	0.057	0.029	0.127	0.071
shef06sem	0.171	0.106	0.126	0.134
shef06ss	0.171	0.106	0.128	0.134

Table 1: Summary of official results from main task submissions.

2.3.3 PhDef

This component, which is used for answering *other* questions (previously referred to as "The Bare Target + Filter + Reduce Approach"), has not changed substantially from that used in the TREC 2005 evaluation (Gaizauskas et al., 2005). For a detailed description of the development of this system see Greenwood (2006).

2.3.4 DefSys

This component, which is used for answering *other* questions (previously referred to as "Target Enrichment + Filter Approach"), has not changed substantially from that used in the TREC 2005 evaluation (Gaizauskas et al., 2005).

3 Run Configurations

The framework described in section 2 allows us to build QA systems by specifying the components to use for the different stages. For example a factoid QA systems consists of an IR engine followed by one or more answer extraction components. In this way we configured two main runs for the main task of the 2006 TREC QA evaluation[3]:

shef06qal This run used QA-LaSIE to answer factoid and list questions using documents retrieved from AQUAINT using Lucene. It uses Lucene and the DefSys component to answer the *other* questions.

shef06sem This run used the semantic tagging approach to answer factoid and list questions using documents retrieved from AQUAINT using Lucene. If this approach fails for list questions then the answers are guessed as described earlier in Section 2.3. It uses Lucene and the PhDef component to answer the *other* questions.

shef06ss This run is identical to shef06sem apart from the fact that factoid and list questions are first attempted using the surface matching patterns approach previously documented by Greenwood and Gaizauskas (2003).

4 Results

The official results from our submitted runs are given in Table 1. It is clear from these results that the semantic tagging approach to answering factoid and list questions greatly outperformed the linguistically motivated QA-LaSIE system. In contrast, the two approaches to answering other questions, PhDef and DefSys, show little difference in performance.

As yet no further analysis of the output of our approaches to question answering with respect to this evaluation have been carried out, and so it would be premature to speculate as to why there is such a difference in performance between the approaches to answering factoid questions. What is clear is that as the two approaches have access to the same information within the documents the semantic tagging approach is much better at selecting the correct answer.

References

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Robert Gaizauskas, Mark A. Greenwood, Mark Hepple, Ian Roberts, Horacio Saggion, and Matthew Sargaison. 2003. The University of Sheffield's TREC 2003 Q&A Experiments. In *Proceedings of the 12th Text REtrieval Conference*.

Robert Gaizauskas, Mark A. Greenwood, Henk Harkema, Mark Hepple, Horacio Saggion, and Atheesh Sanka. 2005. The University of Sheffield's TREC 2005 Q&A Experiments. In *Proceedings of the 14th Text REtrieval Conference*.

[3] A third run was submitted and is also documented here. This run is important only in testing one very small component which affected the answer returned for only four factoid questions in the entire test set, hence the results do not differ significantly from the shef06sem run. Our main aim was to compare the two main approaches to QA, linguistically motivated or shallow semantic tagging, and as such this third run does not contribute to the debate. The difference in the evaluation of other questions between the runs shef06sem and shef06ss is due to differences in the human judgements as the submissions were identical.

Mark A. Greenwood and Robert Gaizauskas. 2003. Using a Named Entity Tagger to Generalise Surface Matching Text Patterns for Question Answering. In *Proceedings of the Workshop on Natural Language Processing for Question Answering (EACL03)*, pages 29–34, Budapest, Hungary, April 14.

Mark A. Greenwood, Ian Roberts, and Robert Gaizauskas. 2002. The University of Sheffield TREC 2002 Q&A System. In *Proceedings of the 11th Text REtrieval Conference*.

Mark A. Greenwood. 2006. *Open-Domain Question Answering*. Ph.D. thesis, The University of Sheffield. Available, October 2006, from `http://www.dcs.shef.ac.uk/~mark/nlp/pubs/`.

Kevin Humphreys, Robert Gaizauskas, Mark Hepple, and Mark Sanderson. 1999. University of Sheffield TREC-8 Q & A System. In *Proceedings of the 8th Text REtrieval Conference*.

Marc Light, Gideon S. Mann, Ellen Riloff, and Eric Breck. 2001. Analysis for Elucidating Current Question Answering Technology. *Natural Language Engineering*, 7(4).

Martin Porter. 1980. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137.

Ian Roberts and Robert Gaizauskas. 2004. Evaluating Passage Retrieval Approaches for Question Answering. In *Proceedings of 26th European Conference on Information Retrieval*.

Sam Scott and Robert Gaizauskas. 2000. University of Sheffield TREC-9 Q & A System. In *Proceedings of the 9th Text REtrieval Conference*.