QED: The Edinburgh TREC-2003 Question Answering System

Jochen L. Leidner   Johan Bos   Tiphaine Dalmas   James R. Curran
Stephen Clark   Colin J. Bannard   Bonnie Webber   Mark Steedman
School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh, EH8 9LW, UK.
trec-qa@inf.ed.ac.uk

Abstract

This report describes a new open-domain answer retrieval system developed at the University of Edinburgh and gives results for the TREC-12 question answering track. Phrasal answers are identified by increasingly narrowing down the search space from a large text collection to a single phrase. The system uses document retrieval, query-based passage segmentation and ranking, semantic analysis from a wide-coverage parser, and a unification-like matching procedure to extract potential answers. A simple Web-based answer validation stage is also applied. The system is based on the Open Agent Architecture and has a parallel design so that multiple questions can be answered simultaneously on a Beowulf cluster.

1 Introduction

This report describes QED, a new question answering (QA) system developed at the University of Edinburgh for TREC-12. A key feature of QED is the use of natural language processing (NLP) technology at all stages in the QA process; recent papers have shown the benefit of using NLP for QA (Moldovan et al., 2002). In particular, we parse both the question and blocks of text potentially containing an answer, producing dependency graphs which are transformed into a fine grained semantic interpretation. A matching phase then determines if a potential answer is present, using the relations in WordNet to constrain the answer.

In order to process very large text collections, the system first uses shallow methods to identify text segments which may contain an answer, and these segments are passed to the parser. The segments are identified using a "tiler", which uses simple heuristics based on the words in the question and the text being processed.

We also use additional state-of-the-art text processing tools, including maximum entropy taggers for POS tagging and named entity (NE) recognition. POS tags and NE-tags are used during the construction of the semantic representation. Section 2 describes each component of the system in detail.

The main characteristics of the system architecture are the use of the Open Agent Architecture (OAA) and a parallel design which allows multiple questions to be answered simultaneously on a Beowulf cluster. The architecture is shown in Figure 1.

2 Component Description

2.1 Pre-processing and Indexing

The ACQUAINT document collection which forms the basis for TREC-2003 was pre-processed with a set of Perl scripts, one per newspaper collection, to identify and normalize meta-information. This meta-information included the document id and paragraph number, the title, publication date and story location. The markup for these last three fields was inconsistent, or even absent, in the various collections, and so collection-specific extraction scripts were required.

The collection was tokenized offline using a combination of the Penn Treebank sed script and Tom Morton's statistical tokenizer, available from the OpenNLP project. Ratnaparkhi's MXTERMINATOR program was used to perform sentence boundary detection (Reynar and Ratnaparkhi, 1997). The result was indexed with the Managing Gigabytes (MG 1.3g) search engine (Witten et al., 1999). For our TREC-2003 experiments, we used case-sensitive indexing without stop-word removal and without stemming.

2.2 Query Generation and Retrieval

Using ranked document retrieval, we obtained the best 100 documents from MG, using a query generated from the question. The question words were

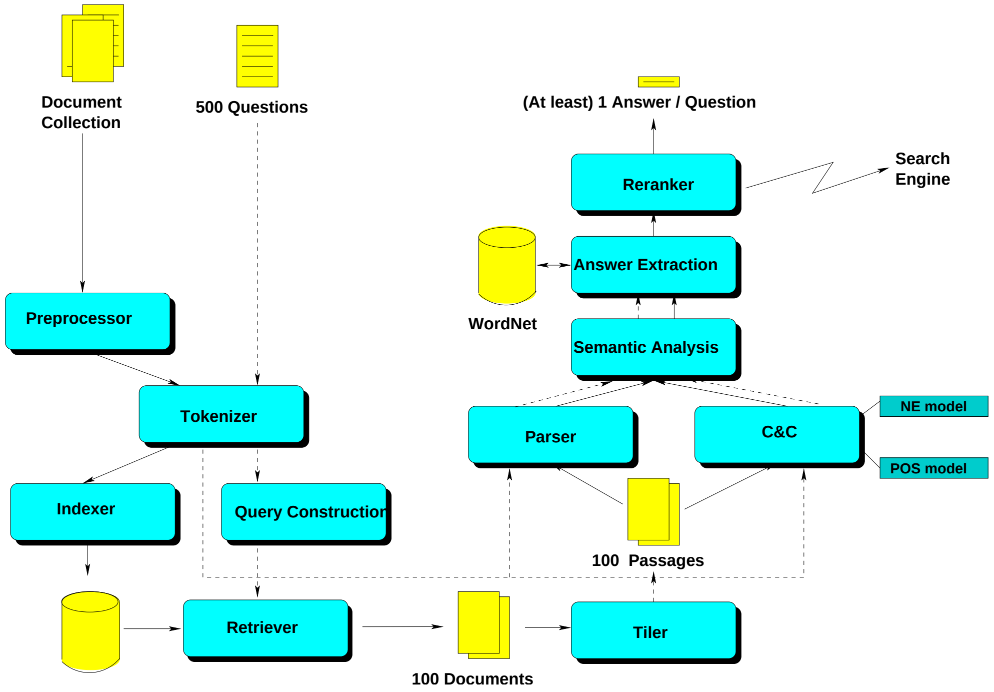

Figure 1: The QED system architecture.

first augmented with their base forms obtained from Minnen et al. (2001)'s morphological analyser, which also performs case normalisation. Stopwords were then removed to form the query keywords. Any remaining uppercase keywords were required to be in the returned documents using MG's plus operator. All remaining lower case keywords were weighted by a factor of 12 (determined by experimentation). Without such a weighting, MG's ranking is too heavily influenced by the uppercase keywords (which are predominantly named entities).

2.3 Passage Segmentation and Ranking

Since our approach involves full parsing to obtain grammatical relations in later stages, we need to reduce the amount of text to be processed to a fraction of the amount returned by the search engine. To this end, we have implemented QTILE, a simple query-based text segmentation and passage ranking tool. This "tiler" extracts from the set of documents a set of segments ("tiles") based on the occurrence of relevant words in a query, which comprises the words of the question. A sliding window is shifted sentence by sentence over the text stream, retaining all window tiles that contain at least one of the words in the query and contain all upper-case query words.

Each tile gets assigned a score based on the following: the number of non-stopword query word tokens (as opposed to types) found in the tile; a comparison of the capitalization of query occurrence and tile occurrence of a term; and the occurrence of 2-grams and 3-grams in both question and tile. The score for every tile is multiplied with a window function (currently a simple triangle function) which weights sentences in the centre of a window higher than in the periphery.

Our tiler is implemented in C++, has linear asymptotic time complexity and requires constant space. For TREC-2003 we use a window size of 3 sentences and pass forward the top-scoring 100 tiles (with duplicates eliminated using a hash signature test).

2.4 Tagging And Syntactic Analysis

The C&C maximum entropy POS tagger (Curran and Clark, 2003a) is used to tag the question words and the text segments returned by the tiler. The C&C NE-tagger (Curran and Clark, 2003b) is also applied to the question and text segments, identifying named entities from the standard MUC-7 data set (locations, organisations, persons, dates, times and monetary amounts). The POS tags and NE-tags are used to construct a semantic representation from the output of the parser (see Section 2.5).

We used the RADISP system (Briscoe and Carroll, 2002) to parse the question and the text segments returned by the tiler. The RADISP parser returns syntactic dependencies represented by grammatical relations such as `ncsubj` (non-clausal subject), `dobj` (direct object), `ncmod` (non-clausal modifier), and so on. The set of dependencies for a sentence are annotated with POS and NE information and converted into a graph in Prolog format. The next section contains an example dependency graph.

To increase the quality of the parser output, we reformulated imperatives in “list questions” (e.g. *Name countries in Europe*) into proper question form (*What are countries in Europe?*). The RADISP parser was much better at returning the correct dependencies for such questions, largely because the RADISP POS tagger typically assigned the incorrect tag to *Name* in the imperative form. We applied a similar approach to other question types not handled well by the parser.

2.5 Semantic Analysis

The aim of this component is to build a semantic representation from the output of the parser. It is used for both the question under consideration and the text passages that might contain an answer to the question. The input to the semantic analysis is a set of dependency relations (describing a graph) between syntactic categories, as Figure 2 illustrates. Categories contain the following information: the surface word-form, the lemmatized word-form, the word position in the sentence, the sentence position in the text, named-entity information, and a POS tag defining the category.

```
top(1, node(’originate’, 9) ).

cat(1, ’croquet’, node(’croquet’, 8), ’NN1’, ’O’ ).
cat(1, ’the’, node(’the’, 5), ’AT’, ’O’ ).
cat(1, ’did’, node(’do’, 4), ’VDD’, ’O’ ).
cat(1, ’originate’, node(’originate’, 9), ’VV0’, ’O’ ).
cat(1, ’game’, node(’game’, 6), ’NN1’, ’O’ ).
cat(1, ’what’, node(’what’, 2), ’DDQ’, ’O’ ).
cat(1, ’country’, node(’country’, 3), ’NN1’, ’O’ ).
cat(1, ’of’, node(’of’, 7), ’IO’, ’O’ ).
cat(1, ’In’, node(’In’, 1), ’II’, ’O’ ).

edge(1, node(’originate’, 9), ncsubj, node(’game’, 6) ).
edge(1, node(’what’, 2), detmod, node(’country’, 3) ).
edge(1, node(’of’, 7), ncmod1, node(’game’, 6) ).
edge(1, node(’of’, 7), ncmod2, node(’croquet’, 8) ).
edge(1, node(’the’, 5), detmod, node(’game’, 6) ).
edge(1, node(’originate’, 9), aux, node(’do’, 4) ).
edge(1, node(’In’, 1), ncmod1, node(’originate’, 9) ).
edge(1, node(’In’, 1), ncmod2, node(’country’, 3) ).

id([’Q_ID’:’1394’,’Q_TYPE’:’factoid’], [1]).
```

Figure 2: Dependency output for the question *In what country did the game of croquet originate?*

Our semantic formalism is based on Discourse Representation Theory (Kamp and Reyle, 1993), but we use an enriched form of Discourse Representation Structure (DRS), combining semantic information with syntactic and sortal information. DRSs are constructed from the dependency relations in a recursive way, starting with an empty DRS at the top node of the dependency graph, and adding semantic information to the DRS as we follow the dependency relations in the graph, using the POS information to decide on the nature of the semantic contribution of a category.

Following DRT, DRSs are defined as ordered pairs of a set of discourse referents and a set of DRS-conditions. The following types of basic DRS-conditions are considered: `pred(x,S)`, `named(x,S)`, `card(x,S)`, `event(e,S)`, and `argN(e,x)`, `rel(x,y,S)`, `mod(x,S)`, where `e`, `x`, `y` are discourse referents, `S` a constant, and `N` a number between 1 and 3. Questions introduce a special DRS-condition of the form `answer(x,T)` for a question type `T`. We call this the *answer literal*; answer literals play an important role in answer extraction (see Section 2.6).

Implemented in Prolog, we reached a recall of around 80%. (By *recall* we mean the percentage of categories that contributed to semantic information in the DRS). Note that each passage or question is translated into one single DRS; hence DRSs can span several sentences. Some basic techniques for pronoun resolution are implemented as well. However, to avoid complicating the answer extraction task too much, we only considered non-recursive DRSs in our TREC-2003 implementation, i.e. DRSs without complex conditions introducing nested DRSs for dealing with negation, disjunction, or universal quantification.

Finally, a set of DRS normalisation rules are applied in a post-processing step, thereby dealing with active-passive alternations, question typing, inferred semantic information, and the disambiguating of noun-noun compounds. The resulting DRS is enriched with information about the original surface word-forms and POS tags, by co-indexing the words, POS tags, the discourse referents, and DRS-conditions (see Figure 3).

```
id([’Q_ID’:’1394’,’Q_TYPE’:factoid],1).

sem(1,

    [p(1001,’In’), p(1002,what), p(1003,country), p(1004,did),
     p(1005,the), p(1006,game), p(1007,of), p(1008,croquet),
     p(1009,originate)],

    [i(1001,’II’), i(1002,’DDQ’), i(1003,’NN1’), i(1004,’VDD’),
     i(1005,’AT’), i(1006,’NN1’), i(1007,’IO’), i(1008,’NN1’),
     i(1009,’VV0’)],

    [drs([0:x1008,1002:x1003,1004:e1004,1005:x1006,1009:e1009],
          [1001:rel(e1009,x1003,’In’),
           1003:answer(x1003,country),
           1006:pred(x1006,game),
           1007:rel(x1006,x1008,of),
           1008:pred(x1008,croquet),
           1009:arg1(e1009,x1006),
           1009:event(e1009,originate) ])]
   ).
```

Figure 3: Example DRS for the question *In what country did the game of croquet originate?*

2.6 Answer Extraction

The answer extraction component takes as input a DRS for the question, and a set of DRSs for selected passages. The task of this component is to extract answer candidates from the passages. This is realised by performing a match between the question-DRS and a passage-DRS, by using a relaxed unification method and a scoring mechanism indicating how well the DRSs match each other.

Taking advantage of Prolog unification, we use Prolog variables for all discourse referents in the question-DRSs, and Prolog atoms in passage-DRSs. We then attempt to unify all terms of the question DRSs with terms in a passage-DRS, using an A* search algorithm. Each potential answer is associated with a score, which we call the DRS score. High scores are obtained for perfect matches (i.e., standard unification) between terms of the question and passage, low scores for less perfect matches (i.e., obtained by “relaxed” unification). Less perfect matches are granted for different semantic types, predicates with different argument order, or terms with symbols that are semantically familiar according to WordNet (Fellbaum, 1998).

After a successful match the answer literal is identified with a particular discourse referent in the passage-DRS. Recall that the DRS-conditions and discourse referents are co-indexed with the surface word-forms of the source passage text. This information is used to generate an answer string, simply by collecting the words that belong to DRS-conditions with discourse referents denoting the answer. Finally, all answer candidates are output in an ordered list. Duplicate answers are eliminated, but answer frequency information is added to each answer in this final list.

Figure 4 gives an example output file. The columns designate the question-id, the source, the ranking score, the DRS score, the frequency of the answer, and a list of sequences of surface word-form, lemma, POS tag and word index.

2.7 Heuristic Candidate Reranking

The system uses a final answer reranking and filtering component, defined slightly differently for each question type. For factoid questions, we rerank the top 5 answers using a function of the candidate answer frequency and the score assigned by the DRS matcher. For definition questions, the same process is used but with a filter which removes any candidate answers with a DRS score below a certain threshold. For list questions, the top 10 answers are considered and the same scoring function is used.

We also use two variations on this reranking algorithm. The first simply uses the DRS score directly, without the candidate answer frequency. The second uses frequency counts from Google to filter out improbable question-answer combinations. A query is sent to Google based on a combination of keywords from the question and the candidate answer. If the document count returned by Google is below some threshold, the answer candidate is removed.

3 Evaluation

Three runs were submitted: run A (EdinInf2003A) used Google as a filter; run B is the system using a function of the candidate answer frequency and the DRS score; and run C is the system using the DRS score directly. On factoid questions, we obtained an accuracy of 0.073 (372 wrong, 5 correct but unsupported, 6 inexact, 30 correct) for runs A and B. For run C we obtained a score of 0.058. Figure 5 gives some example extracted answers for factoid questions.

```
What country is Aswan High Dam located in?
R 1900 XIE19960828.0011 Egypt
What business was the source of John D. Rockefeller’s fortune?
R 1909 NYT19991109.0441 Standard Oil
How many Earth days does it take for Mars to orbit the sun?
R 2000 NYT19991220.0063 687
What river is under New York’s George Washington bridge?
R 2330 NYT20000203.0416 the Hudson River
What instrument did Louis Armstrong play?
R 2356 NYT19990830.0439 trumpet
What is the name of the Chief Justice of the Supreme Court?
R 2198 XIE19971101.0185 Sajjad Ali Shah
What membrane controls the amount of light entering the eye?
R 1941 APW19980609.1138 The iris
What museum in Philadelphia was used in ”Rocky”?
R 2044 NYT20000411.0123 the Museum of Art
When was the first Star Wars movie made?
R 2069 NYT19990315.0214 1977
What composer wrote ”Die Gotterdammerung”?
R 2301 NYT19980629.0183 Wagner

How late will airlines let you fly in pregnancy?
W 1952 NYT19990101.0001 NIL
How fast can a nuclear submarine travel?
W 1937 APW20000814.0076 24 nuclear armed
                        cruise missiles
How many floors are in the Empire State Building?
W 1938 NYT19990121.0328 only the top
                        22 floors
What did George Washington call his house?
W 1944 APW19990728.0148 the picture of
                        George Washington
```

Figure 5: Some correct (R) and wrong (W) answers from the EdinInf2003A run. The second column of the system response contains the question number; the third column contains the document the answer was retrieved from.

See Figure 6 for a breakdown of factoid questions by *wh*-word for runs A and B. We obtained correct, but unsupported, answers for factoid questions 1971, 2023, 2048, 2115, 2245 in runs A and B and a similar list excluding the latter two questions for run C. Our average F score over 37 list questions was 0.013; for the 50 definition questions we obtained an F score of 0.063. As a result, our final main task score is 0.056.

4 Discussion and Future Work

In TREC 2003, the overall accuracy of the 54 runs submitted to the QA track ranged between 0.034 and 0.700

```
1394 NYT19990821.0176 0.0687983 0.50 8 Degnan Degnan NNP 157001
1394 NYT19990821.0176 0.0687983 0.43 3 the the DT 158010 nation nation NN 158011
1394 APW19990616.0182 0.0923594 0.37 1 Tarzan Tarzan NNP 21011
1394 APW20000827.0133 0.0651768 0.37 2 English English NN 219015
1394 APW20000827.0133 0.0651768 0.37 1 Additionally Additionally NNP 220001
1394 APW20000827.0133 0.0651768 0.37 4 the the DT 220010 U.S. U.S. NNP 220011
```

Figure 4: Example output file of answer extraction.

WHAT	25/230
WHAT + LOCTYPE	16
WHAT + BE	2
WHAT [OTHER]	7
WHEN	4 / 39
HOW + ADV	1 / 100

Figure 6: Breakdown of correct factoid answers by *wh*-word.

(median 0.177). For list questions, the best, median, and worst average F-scores were 0.396, 0.069, and 0.000, respectively. For definition questions, the F-scores ranged from 0 to 0.555 (with a median of 0.192).

In relation to this interval, our low score reflects the fact that our first year of track QA participation required a large resource commitment to develop a solid basic infrastructure. Such long-term investment will provide the basis for subsequent performance analysis, which in turn will lead to replaced components with superior performance.

Acknowledgements

We are grateful Malvina Nissim for advice on Google and RASP and to the authors of all external progams we utilized for making them available. Leidner is supported by the German Academic Exchange Service (DAAD) under scholarship D/02/01831 and by Linguit GmbH (research contract UK-2002/2). Dalmas is supported by the School of Informatics, University of Edinburgh. Curran is supported by a Commonwealth scholarship and a Sydney University Travelling scholarship. Clark is supported by EPSRC grant GR/M96889. Bannard is supported by ESRC Grant PTA-030-2002-01740.

References

[Briscoe and Carroll2002] Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, Las Palmas, Gran Canaria.

[Curran and Clark2003a] James R. Curran and Stephen Clark. 2003a. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 11th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 91–98, Budapest, Hungary.

[Curran and Clark2003b] James R. Curran and Stephen Clark. 2003b. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada.

[Fellbaum1998] Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

[Kamp and Reyle1993] Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

[Minnen et al.2001] Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Journal of Natural Language Engineering*, 7:207–223.

[Moldovan et al.2002] Dan Moldovan, Sanda Harabagiu, Roxana Girju, Paul Moraarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. 2002. LCC tools for Question Answering. In *Proceedings of the Eleventh Text Retrieval Conference (TREC-2002)*, Gaithersburg, Maryland.

[Reynar and Ratnaparkhi1997] Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C.

[Witten et al.1999] Ian A. Witten, Alistair Moffat, and Timothy C. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, Los Altos, CA, 2nd edition.