University of Sheffield TREC-8 Q & A System

Kevin Humphreys[a] Robert Gaizauskas[a]
Mark Hepple[a] Mark Sanderson[b]
{k.humphreys,r.gaizauskas,m.hepple}@dcs.shef.ac.uk
m.sanderson@shef.ac.uk

[a]Department of Computer Science / [b]Department of Information Studies
University of Sheffield
Regent Court, Portobello Road
Sheffield S1 4DP UK

1 Introduction

The system entered by the University of Sheffield in the question answering track of TREC-8 is the result of coupling two existing technologies – information retrieval (IR) and information extraction (IE). In essence the approach is this: the IR system treats the question as a query and returns a set of top ranked documents or passages; the IE system uses NLP techniques to parse the question, analyse the top ranked documents or passages returned by the IR system, and instantiate a query variable in the semantic representation of the question against the semantic representation of the analysed documents or passages. Thus, while the IE system by no means attempts "full text understanding", this approach is a relatively deep approach which attempts to work with meaning representations.

Since the information retrieval systems we used were not our own (AT&T and UMass) and were used more or less "off the shelf", this paper concentrates on describing the modifications made to our existing information extraction system to allow it to participate in the Q & A task.

2 System Description

2.1 Overview

The key features of the system setup are shown in Figure 1. Firstly, the TREC document collection and each question were passed to two IR systems which treated the question as a query and returned top ranked documents or passages from the collection. As one IR system we used the AT&T supplied top documents which were made available to all participants by NIST; as the second we used the passage retrieval facilities of the University of Massachusetts Inquery system [2] to return top ranked passages. Following this, for each question, the question itself and the top ranked documents or passages were processed by a slightly modified version of the LaSIE information extraction system [7], which we refer to below as QA-LaSIE. This yielded two sets of results which were entered separately for the evaluation – one corresponding to each of the IR systems used to filter the initial document collection.

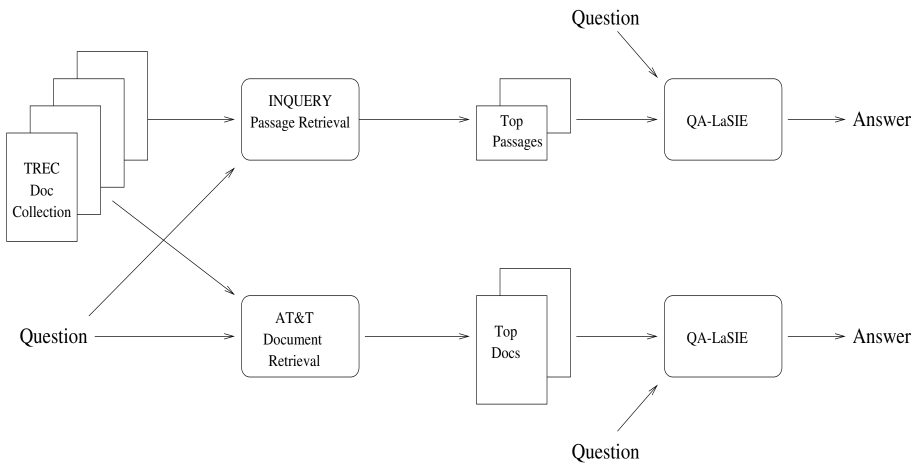

Figure 1: System Setup for the Q & A Task

The reasoning behind this choice of architecture is straightforward. The IE system can perform detailed linguistic analysis, but is quite slow and could not process the entire TREC collection for each query, or even realistically preprocess it in advance to allow for reasonable question answering performance during the test run. IR systems on the other hand are designed to process huge amounts of data. Thus, the hope was that by using an IR system as a filter to an IE system we could benefit from the strengths of each [6].

In the next section we describe the basic LaSIE system and then in succeeding sections proceed to describe the modifications made to it for the TREC-8 Q & A task.

2.2 LaSIE

The LaSIE system used to perform the detailed question and text analysis is largely unchanged from the IE system as entered in the most recent Message Understanding Conference evaluation (MUC-7) evaluation [7]. The principal components of the system are shown in Figure 2 as executed interactively through the GATE Graphical Interface [4]. The system is essentially a pipeline of modules each of which processes the entire text before the next is invoked. The following is a brief description of each of the component modules in the system:

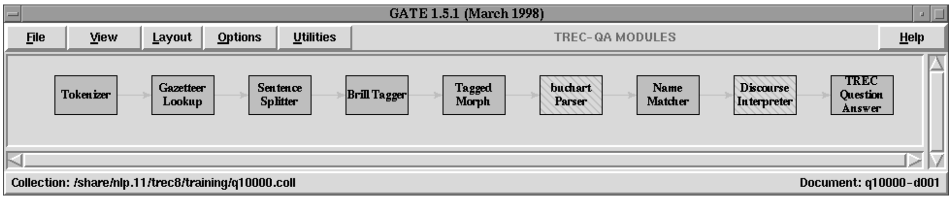

Figure 2: QA-LaSIE System Modules

Tokenizer Identifies token boundaries (as byte offsets into the text) and text section boundaries (text header, text body and any sections to be excluded from processing).

Gazetteer Lookup Identifies single and multi-word matches against multiple domain specific full name (locations, organisations, etc.) and keyword (company designators, person first names, etc.) lists, and tags matching phrases with appropriate name categories.

Sentence Splitter Identifies sentence boundaries in the text body.

Brill Tagger [1] Assigns one of the 48 Penn TreeBank part-of-speech tags to each token in the text.

Tagged Morph Simple morphological analysis to identify the root form and inflectional suffix for tokens which have been tagged as noun or verb.

Parser Performs two pass bottom-up chart parsing, pass one with a special named entity grammar, and pass two with a general phrasal grammar. A 'best parse' is then selected, which may be only a partial parse, and a predicate-argument representation, or quasi-logical form (QLF), of each sentence is constructed compositionally.

Name Matcher Matches variants of named entities across the text.

Discourse Interpreter Adds the QLF representation to a semantic net, which encodes the system's world and domain knowledge as a hierarchy of concepts. Additional information inferred from the input is also added to the model, and coreference resolution is attempted between instances mentioned in the text, producing an updated discourse model. A representation of the question is then matched against the model, using the coreference mechanism.

Question Answer Selects the required answer text using the resolved question representation in the discourse model.

2.3 QA-LaSIE

The QA-LaSIE system operates by processing an ordered set of texts for each question with the question itself as the first text. The IR systems' results were split into a subdirectory for each question, containing, firstly, the question itself, then, in rank order, a predefined number of texts or passages retrieved for that question. For the Inquery data, the top 10 passages were used, and for the AT&T data, the top 5 full texts. These limits were chosen mainly to restrict the system's total processing time, but for the Inquery data the limit was based on a partial analysis of the rankings of texts containing a correct answer for the training set of questions.

For the evaluation, QA-LaSIE was run in batch mode to process each subdirectory of question plus retrieved texts. When an answer was found, 50- and 250-byte responses were written out, and processing moved immediately to the next question, as described below. The system required an average of around 15 minutes to process each question and its corresponding set of retrieved texts on a SUN Sparc 5 machine, though no effort has been spent on optimisation.

The following subsections detail the modifications required for the original IE system to operate in a question answering mode.

2.3.1 Question Parsing

An additional subgrammar was added to the phrasal parsing stage for interrogative constructions, which were not handled at all in the original LaSIE system. The grammar was developed until reasonable coverage on the 37 questions in the training set was obtained, with only a very limited attempt to cover constructions outside this set. Compositional semantic rules on each syntactic rule are used to build up a 'quasi-logical form' (QLF) representation, in the same way as the rest of the grammar. A special semantic predicate, `qvar` (question variable), is used in the semantics to indicate the 'entity' requested by the question. For example, the question *Who composed Eugene Onegin?* would produce the following QLF representation:

```
qvar(e1), person(e1)
name(e2,'Eugene Onegin')
compose(e3), tense(e3,past)
lsubj(e3,e1), lobj(e3,e2)
```

Here, each entity in the question gives rise to a unique identifier of the form en. The use of the lexical item *who* causes the addition of `person(e1)`, but the semantic class of `e2` (*Eugene Onegin*) is unspecified. The relational predicates `lsubj` (logical subject) and `lobj` (logical object) simply link any verb arguments found in the text, rather than using any subcategorisation information to determine the arguments required for a particular verb.

The QLF representation of each question is stored for use in the subsequent processing of each candidate answer text. After parsing, the question is processed by the Namematcher and Discourse Interpreter modules, but the results of these modules are currently unused. Potentially, these modules could carry out coreference resolution within the question, thus allowing complex, even multi-sentence, questions to be processed, but this capability was not required for any of the questions in the training set and was not used for the test run.

2.3.2 Question Resolution

The candidate texts for each question are processed exactly as in the standard LaSIE system, up until the completion of the Discourse Interpreter stage. At this point, if a stored representation of a question for the current text is found, this representation is examined and an attempt made to find an answer within the text's completed discourse model. Each question representation gives rise to a hypothesised entity (the `qvar`), and the Discourse Interpreter's general coreference mechanism is used to attempt to find an 'antecedent' for the hypothesis from the text.

Various restrictions are placed on the hypothesised entity from the question's QLF representation. The entity required to answer the question will be flagged as having the semantic class `qvar`, but it may also have other semantic types, such as `person` if the question introduces the entity using *Who*, as in the example above. The entity may also have other attributes mentioned in the question, such as `name`, and attributes linking the `qvar` entity to other entities from the question, in particular the verb argument relations `lsubj` and `lobj`.

In some cases the question grammar may fail to parse a question as an interrogative construction, and the parser will produce only a partial QLF representation which does not include a `qvar`. In this case the discourse interpreter applies a fallback mechanism to force the first text in each question/answer set to be interpreted as a question, simply treating the first entity in a QLF representation with no `qvar` as the `qvar`. The first entity is currently chosen arbitrarily, with no analysis of the partial QLF representation, but the mechanism does allow the system to recover from the incomplete coverage of the question grammar, and still produce answers even where no question was recognised.

Anaphor Resolution Before attempting to resolve the `qvar` entity, the general coreference mechanism is applied to any other entities from the question. The coreference mechanism currently only attempts to resolve the classes of anaphora defined for the MUC-7 evaluation, i.e. identity relations between proper names, pronouns, noun phrase heads and noun modifiers. No general attempt is currently made to resolve multiple descriptions of events in a text, though this is attempted for question resolution, as described below.

The general coreference mechanism, described fully in [5], acts to compare pairs of entities to determine a similarity measure. Firstly, the semantic classes of the two entities are compared (semantic type compatibility) by testing for a dominance relation within the system's ontology, or concept hierarchy. Secondly, if the semantic classes are compatible, the values of all 'immutable' (fixed single-valued) attributes (e.g. `gender`, `number`) are compared (attribute similarity) to ensure no conflicts exist. Thirdly, an overall similarity score is calculated, combining the distance between the semantic classes of the two instances, and the number of shared, non-immutable attributes.

For each potential anaphor, if any comparison pairs are assigned a similarity score, the entity with the highest score will be merged with the anaphor in the discourse model. This results in the representation of a single entity in the discourse model which has multiple realisations in the text, i.e. a coreferential entity.

Event Similarity For hypothesised `qvar` question answer entities, an additional, fourth, comparison stage has been added to the coreference mechanism to ensure that a candidate antecedent, or answer, shares any relations to `event` entities (`lsubj`, `lobj` or `comp` (complement)). This is required to allow the resolution of the `qvar` from a question like *Who composed Eugene Onegin* with an entity from a text containing *Tchaikovsky wrote Eugene Onegin*. The `qvar` entity here is the logical subject of the `compose` event, but to resolve this with *Tchaikovsky*, the candidate antecedent must have a `lsubj` relation with an event of a compatible class and with the same arguments, `lobj` in this case, via coreference between the question and the text.

This additional stage therefore requires the identification of events of compatible classes, testing semantic type similarity within the system's ontology. However, rather than explicitly extending the ontology to include as many concepts as possible, and introducing all the problems of word sense ambiguity, a simple high-level general ontology was defined, and then reference made to WordNet [3] hypernym/hyponym relations during processing. When attempting to find an antecedent for the `qvar` above, the `compose` event would be compared with the `write` event using the relations between WordNet synsets. An arbitrary limit of 3 hypernym/hyponym links was used to constrain the event similarity test, and, in this case, only a single link is required in WordNet to relate *compose* and *write*. The distance between the two event classes is then combined with the general coreference mechanism's similarity score for the `qvar` antecedent, so preferring antecedents which are arguments of more similar event classes.

The copular verb *be* was treated specially when comparing it to other event classes. The grammar treats the copular as any other verb, introducing an event instance for it, but in the event similarity test it is treated as being compatible with any other event class, though with a low score.

The general approach to ontology construction in the LaSIE system has previously been to only include concepts directly relevant to a particular IE task. The tasks have been fixed and well defined, so a small domain-specific ontology has been sufficient. For the Q & A task, however, no assumptions about the domain of each question can be made, and so a more general purpose ontology is required. Reference to the WordNet hierarchy is currently only made for comparing event classes. A similar comparison could also be made for object classes, effectively extending the system's object hierarchy as necessary, but this was not implemented for the Q & A evaluation.

2.3.3 Answer Generation

An additional Q & A task-specific module was added to the LaSIE system, following the Discourse Interpreter stage. This module simply scanned the final discourse model for each text to check for an instantiated `qvar`, i.e. a `qvar` that had been successfully resolved with an entity in the text. If found, the realisation of that entity in the text (the longest in the case of multiple realisations via coreference resolution) was used as the central point from which 50- and 250-byte text windows were extracted to be used as question responses.

A significant feature of the QA-LaSIE system's operation is that once a response for a particular question has been produced, no further candidate texts are processed for that question. This was partly to improve system performance by avoiding any unnecessary processing of texts once an answer had been produced. However, this did assume that the IR systems' ranking of the candidate texts was accurate. The highest ranked text was processed first, and if an answer was produced from it, lower ranked texts were not considered. With hindsight, this approach was really at odds with the Q & A task's intended mode of operation, where multiple ranked answers for each question were expected. The QA-LaSIE system could easily be adapted to return multiple answers, and re-use the IR systems' rankings, but the single-answer mode reflects the original IE approach.

3 Results and Analysis

Since the QA-LaSIE system only ever produced a single answer for each question, which was arbitrarily assigned a ranking of 1, the official results evaluating the accuracy of system rankings are not particularly meaningful for QA-LaSIE [1]. Therefore, an initial analysis of the system results has been carried out to attempt to express performance in the standard recall and precision metrics (in this context recall is the proportion of questions correctly answered, precision the proportion of answered questions for which the answer is correct).

The following results were obtained from the individual judgements of question answers and an analysis of the system's intermediate outputs for each question.

For the NIST-supplied AT&T data, where the top 5 full texts for each question were processed, the overall results were:

```
50-byte answers:                      250-byte answers:
Recall    = 14 / 198 =  7.07%         Recall    = 19 / 198 =  9.59%
Precision = 14 / 60  = 23.33%         Precision = 19 / 60  = 31.67%
```

For the University of Massachusetts Inquery data, where the top 10 passages for each question were processed, the overall results were:

```
50-byte answers:                      250-byte answers:
Recall    = 16 / 198 =  8.08%         Recall    = 22 / 198 = 11.11%
Precision = 16 / 61  = 26.23%         Precision = 22 / 61  = 36.06%
```

A more detailed analysis of the QA-LaSIE results alone, separate from the retrieval system, was then carried out. This involved attempting to determine,

[1] The adjudicated mean reciprocal rank scores were as follows. For 250-byte answers, .111 for the Inquery supplied top 10 passages, .096 for the AT&T supplied top 5 full texts. For 50-byte answers, .081 for the Inquery data, .071 for the AT&T supplied data.

for each question, whether the retrieval results used did in fact include a text containing an answer. To avoid manually judging every text, the Q & A task judgements of all system results were used. The definition of a correctly retrieved text is therefore a text from which any system in the evaluation produced a correctly judged answer, though clearly there may be other retrieved texts which also contain answers. Using this definition, the top 5 texts from the AT&T data represented 71.72% recall of correct question answers, and the top 10 passages from the Inquery data represented 76.26% recall (though no manual test has been done to ensure the correct passages were selected from the texts).

Analysing the QA-LaSIE results for only those questions for which texts were correctly retrieved produced the following figures for the AT&T data:

```
50-byte answers:                      250-byte answers:
Recall    = 14 / 141 =  9.87%         Recall    = 19 / 141 = 13.38%
Precision = 14 / 47  = 29.79%         Precision = 19 / 47  = 40.43%
```

and for the Inquery data:

```
50-byte answers:                      250-byte answers:
Recall    = 16 / 151 = 10.60%         Recall    = 22 / 151 = 14.57%
Precision = 16 / 49  = 32.65%         Precision = 22 / 49  = 44.90%
```

A further analysis considered system performance for only those questions which were parsed as interrogative constructions (i.e. where the QLF representation included a `qvar`), and where texts containing an answer were correctly retrieved. This excludes some cases where the system produced answers, some correct, despite the QLF representation of the question containing no `qvar`, using the fallback mechanism described in Section 2.3.2. For the AT&T data, the results are:

```
50-byte answers:                      250-byte answers:
Recall    = 13 / 87 = 14.94%          Recall    = 17 / 87 = 19.54%
Precision = 13 / 42 = 30.95%          Precision = 17 / 42 = 40.48%
```

and for the Inquery data:

```
50-byte answers:                      250-byte answers:
Recall    = 12 / 84 = 14.28%          Recall    = 18 / 84 = 21.43%
Precision = 12 / 40 = 30.00%          Precision = 18 / 40 = 45.00%
```

These results give a better indication of the performance of the QA-LaSIE system alone, attempting to exclude the particular IR system used, and also the current incomplete state of the question grammar.

4 Conclusion

We have not yet carried out detailed failure analysis of the QA-LaSIE system, and so cannot make many specific claims about where the strengths and weaknesses of the approach lie. The performance of the system clearly leaves much to be desired, and the results are very low using the reciprocal ranking measure used in the evaluation. However this measure, and indeed the current Q & A task methodology, does not allow any useful measure of precision. Answers to all test questions are known to be available in the test corpus, and the ability to return negative results where appropriate is not evaluated.

Given the very limited effort that went into tuning the QA-LaSIE system we believe that the approach performed sufficiently well to warrant further investigation. The system was assembled in less than two person weeks, and very little effort was available to adapt the general coreference mechanism to the task of question resolution.

Several areas where further work or investigation are clearly needed are:

- *Question Parsing* As only 2/3 of the questions were parsed more effort is needed to refine and extend the coverage of the question grammar.

- *Answer Text Processing* Analysis needs to be carried out to see to what extent the meaning representations computed for the answer texts do or do not contain the information required to answer the questions. If not, the source of this inadequacy needs to be identified (faulty parsing, inadequate lexical or world knowledge).

- *QVAR Coreference* Analysis of whether the `qvar` matching in the coreference mechanism is too weak or too strong needs to be carried out. Strict insistence that all attributes associated with the `qvar` in the question be matched in a candidate answer text may be too strong a requirement; on the other hand loosening the match may result in spurious answers.

- *General Purpose Ontology* The ontology used in the QA-LaSIE system, while intended to be general purpose, is actually abstracted from a small number of business domains used in the development of the LaSIE IE system. This clearly has only a very limited coverage of the varied domains represented in an unconstrained set of questions. Considerable further investigation into ways of extending the coverage is required, including evaluation of the use of available resources such as WordNet, as implemented here for event classes.

- *Multiple Answers* As noted, QA-LaSIE halts after returning the first answer it finds for each question. It would be relatively trivial to extend the system to process all the documents passed to it by the IR system and rank the resulting answers. The impact of this on performance needs to be assessed.

Such investigations will help to reveal whether the approach we have followed for the Q & A task is appropriate. More generally they will shed light on the very interesting questions this task throws up: to what extent are 'deeper' models of language processing necessary to perform a question answering task against large text collections.

Acknowledgements

The authors would like to thank James Allan and Daniella Malin of the Computer Science Department, University of Massachussetts for supplying the results of running the Inquery system with the Q & A questions as queries against the TREC data collection.

References

[1] E. Brill. A simple rule-based part-of-speech tagger. In *Proceeding of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy, 1992.

[2] J.P Callan, W.B. Croft, and S.M. Harding. The INQUERY retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert System Applications*, pages 78–83, 1992.

[3] Miller G.A., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: On-line. Distributed with the WordNet Software., 1993.

[4] R. Gaizauskas, H. Cunningham, Y. Wilks, P. Rodgers, and K. Humphreys. GATE – an Environment to Support Research and Development in Natural Language Engineering. In *Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-96)*, pages 58–66, Toulouse, France, October 1996.

[5] R. Gaizauskas and K. Humphreys. Quantitative Evaluation of Coreference Algorithms in an Information Extraction System. In S. Botley and T. McEnery, editors, *Discourse Anaphora and Anaphor Resolution*. Forthcoming. Also available as Department of Computer Science, University of Sheffield, Research Memorandum CS – 97 – 19, `http://www.dcs.shef.ac.uk/research/resmems`.

[6] R. Gaizauskas and A.M. Robertson. Coupling information retrieval and information extraction: A new text technology for gathering information from the web. In *Proceedings of the 5th Computed-Assisted Information Searching on Internet Conference (RIAO'97)*, pages 356–370, Montreal, 1997.

[7] K. Humphreys, R. Gaizauskas, S. Azzam, C Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. Description of the LaSIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.