A Combined IR/NLP Approach to Question Answering Against Large Text Collections

Robert Gaizauskas and Kevin Humphreys
Department of Computer Science, University of Sheffield
Regent Court, Portobello Street
Sheffield S1 4DP UK
{r.gaizauskas,k.humphreys}@dcs.shef.ac.uk

Abstract

We describe an approach to finding literal answer strings to natural language questions in large text collections. The approach involves linking an IR system with an NLP system that performs reasonably thorough linguistic analysis. The IR system treats the question as a query and returns a set of top ranked documents or passages. The NLP system parses the question and analyses the top ranked documents or passages returned by the IR system, yielding a 'meaning representation' of each. It then instantiates a privileged query variable in the semantic representation of the question against the semantic representation of the analysed documents or passages to discover the answer, using a general purpose coreference mechanism. The approach has been evaluated in the TREC-8 question and answer (Q & A) track evaluation. While initial overall success is limited, it is sufficient to warrant further investigation of the approach. In particular this work will shed light on the interesting question that the TREC Q & A task poses: to what extent are 'deeper' models of language processing necessary to perform question answering against large text collections?

Introduction

Traditionally, information retrieval (IR) systems are conceived as systems whose purpose is to return relevant documents in response to a user query. However, such systems are more accurately termed *document* retrieval systems than *information* retrieval systems. For, once the documents are returned, the user must carry out the additional step of reading the documents returned by the system and extracting the information he or she seeks from them. If the number of documents returned is small and they are not too long and the user's information requirement is general rather than specific, then this behaviour may be entirely acceptable. But, if there are large numbers of documents or they are of considerable length or the information sought is quite specific, then this step of extracting information from returned documents may become unacceptably burdensome. In particular, if the user seeks the answer to a specific question such as *What date in 1989 did East Germany open the Berlin Wall?* and has at hand a text collection of several gigabytes in which the answer may be presumed to be found, then clearly they would prefer a response such as *November 9*, perhaps with some small amount of context (e.g. a sentence), to a (ranked) set of documents which they must read to discover the answer.

It was with the aim of encouraging research into question answering systems of this sort that the eighth Text Retrieval Conference (TREC-8) introduced a question answering (Q & A) track for the first time. Details of this task are supplied below, but in essence the task required a system to find literal answer strings for a set of single sentence questions from a large collection of short texts (newswire articles). Of course research into question answering is not new. Leaving aside the long tradition of work in AI on deductive question answering, which addresses questions of how answers to logical queries may be derived from logic databases (see, e.g., Green (1969), Schubert and Watanabe (1986)), there has

been much work by the NL community on various aspects of natural language question answering. Much, though by no means all of this work has centered on natural language front ends to databases (see, e.g. Copestake and Jones (1990) for a review). However, there are a number of features of the TREC Q & A task that make it distinct from previous work on question answering.

1. Like the NL front-end to database task, the questions to be answered are in potentially unrestricted NL (though they must be a single sentence and may not be clarified through a dialogue between the system and user); however, unlike this task the repository of information from which the question is to be answered is represented not as a structured database, but as a collection of unstructured natural language texts.

2. The text collection in which the answer must be found is very large – two volumes of the TREC text research collection, comprising nearly a gigabyte of newswire text.

3. Unlike the deductive question answering task, where answers may be logical consequences of facts or rules stored in the database, and hence are *implicit*, in the Q & A task answers are always *explicit* – actual strings from the source texts. However, this does not mean that inference, or something akin to it, is not necessary in the question answering process – coreference resolution, part-whole reasoning, lexical semantic knowledge and world knowledge may all potentially be required to answer the questions, since the form of the question may bear no relation to the sentence containing the answer.

The Q & A task therefore poses a new and challenging task for information retrieving systems. It also raises, once more, the issue of to what extent natural language processing techniques may or may not contribute to a solution. There has been much discussion of the utility of NL techniques for the document retrieval task (Smeaton, 1999; T.Strzalkowski et al., 1999) with many researchers concluding that there is in fact little they can contribute that cannot be achieved by other, simpler means (Sparck Jones, 1999). However, even sceptics like Sparck Jones suggest that there may be a role for NLP techniques in question answering:

> But it is clear that selecting key information and using it to form an information base, for future question-answering, in general depends on linguistic analysis, even if this may sometimes be done by linguistically shallow means. (Sparck Jones (1999), p. 22)

She is here referring specifically to information extraction systems of the sort developed in response to the MUC programme (e.g. Def (1995)), whose function is to construct a structured database of pre-defined form by extracting information from unstructured texts, the database then being available for querying using database query techniques. But it seems reasonable to anticipate the same role for NL techniques in the Q & A task. However, this is an open question and one which the TREC Q & A track will contribute to answering: once again it may prove that non-linguistically motivated techniques will prove simpler and perhaps superior to linguistically motivated ones.

The present paper is a contribution to this debate. We describe a system entered by the University of Sheffield into the TREC-8 Q & A track. This system 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", our approach is a relatively deep one, which attempts to work with meaning representations.

The paper is structured as follows. The first section below offers motivation for why a NL approach to question answering may be worth pursuing. The next section describes our question answering system, first giving an overview, then discussing the LaSIE information extraction system in general and as modified to perform the question answering task. Since the information retrieval systems we used were not our own and were used more or less "off the shelf", we concentrate on describing the modifications made to our existing information extraction system to allow it to participate in the Q & A task. The following section describes the experimental setup of the TREC-8 Q & A task in more detail, sufficient for the results and analysis presented in the penultimate section to make sense. We conclude with a general assessment of our approach and observations about how to take it forward, with some discussion of related work at TREC-8 and with some reflections on the TREC-8 Q & A task and its relation to question answering viewed more generally.

The Potential of NLP for Question Answering

While the system we describe below uses both IR and NLP techniques, its distinctiveness lies in attempting to carry out a linguistically motivated analysis of both question and text. In this section we briefly present a number of examples which illustrate why NLP techniques can in principle aid in question answering. As noted, whether they can in practice remains to be shown.

Dealing with the following linguistic phenomena may be critical in question answering:

1. *Coreference* Part of the information required to answer a question may occur in one sentence, while the rest occurs in another – potentially at some distance, linked to the first via a pronominal or other anaphoric link. For example, to answer the question *How much did Mercury spend on advertising in 1993?* given a text which ends with the sentence *Last year the company spent Pounds 12m on advertising.* requires coreferring the definite noun phrase *the company* back to *Mercury*, which in this (real) example was last mentioned three sentences earlier.

2. *Deixis* News texts are written at a particular time and frequently from a particular place, and deixis in the text needs to be interpreted in this light. For example, to interpret *last year* as 1993 in the preceding example, the year specified in the question, requires processing the date-line for this text (1994) and correctly dereferencing the deitic expression.

3. *Grammatical knowledge* Difference in grammatical role can be of crucial importance. A question such as *Which company took over Microsoft?* (artificial example), where Microsoft appears as the grammatical object of the verb will not be answered by sentences in which Microsoft occurs as the subject of a take-over, e.g. *Microsoft took over Entropic*.

4. *Semantic knowledge* Consider the question *At what age did Rossini stop writing opera?* and the text *Rossini . . . composed both The Barber of Seville and La Cenerentola before he was 25 and he did not write another opera after he was 35*. To answer this question requires realising that not writing another opera after 35 implies stopping by age 35.

5. *World knowledge* Consider the question *How much did Manchester United spend on players in 1993?*. An answer to this can be found in a text which requires knowing that the nick-name of Manchester United is *the Reds*.

This list by no means exhausts those linguistic phenomena which may need to be taken into account to accurately answer questions. But it should serve to motivate the use of a system which attempts to model these phenomena in the question answering task. The system we describe in the next section does attempt such modeling, albeit in a limited fashion.

System Description

Overview

The key features of the system setup as used in the TREC-8 Q & A task 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, the TREC-8 organisers; as the second we used the passage retrieval facilities of the University of Massachusetts Inquery system (Callan et al., 1992) 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 (Humphreys et al., 1998), 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.

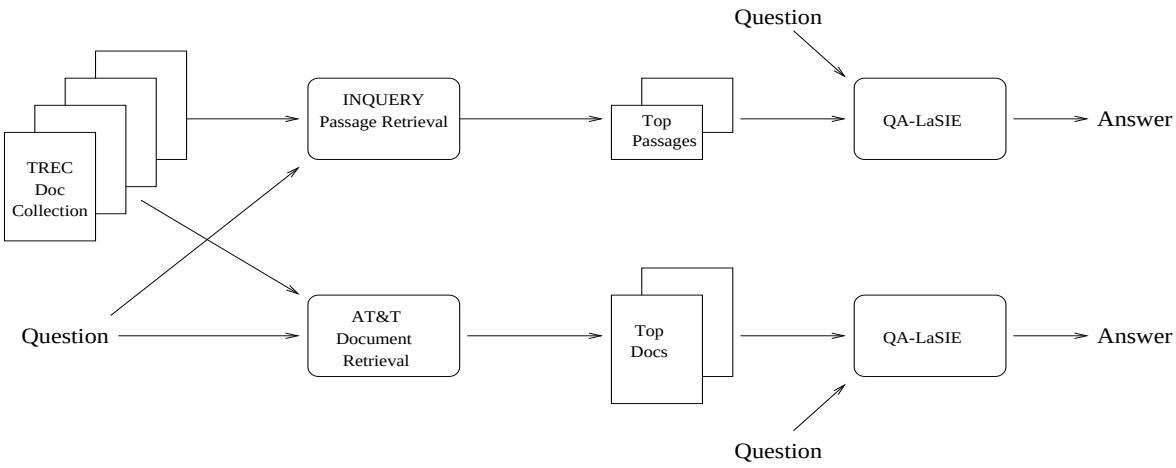

Figure 1: System Setup for the Q & A Task

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 pre-process 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 strategy was to use an IR system as a filter to an IE system, allowing us to benefit from the strengths of each (for an alternative exploitation of this setup see Gaizauskas and Robertson (1997)).

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.

LaSIE

The LaSIE system used to perform the detailed question and text analysis is largely unchanged from the system as entered in the MUC-7 evaluation (Humphreys et al., 1998). 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:

- *Tokenizer* Identifies token boundaries 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* (Brill, 1992) Assigns one of the 48 Penn TreeBank part-of-speech tags to each token in the text.
- *Tagged Morph* Performs 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 of each sentence, pass one with a special named entity grammar, and pass two with a general phrasal grammar. The named entity grammar (341 rules) identifies expressions falling into the MUC-7 named entity classes of person, organization, location, date, time, percent and monetary amounts. The general phrasal grammar (148 rules) reliably recognises a restricted class of noun phrases, verb groups, prepositional phrases, relative clauses, and sentences, but is conservative about performing attachment, leaving phrases unattached in cases of ambiguity. From the chart a 'best parse' is selected, which may consist of multiple phrasal fragments in case no single spanning analysis has been found, and a predicate-argument representation, or quasi-logical form (QLF), of each fragment is constructed compositionally, the set of these QLF's becoming the semantic representation of the sentence.
- *Name Matcher* Matches variants of named entities across the text (e.g. *Ford Motor Company* and *Ford*).
- *Discourse Interpreter* Successively integrates the QLF representation of each sentence into a semantic net which encodes the system's world knowledge as a hierarchy of concepts and models the content of the current discourse as a specialisation of this knowledge. As each QLF representation is added, additional information presupposed by it is also added to the model; then coreference resolution is performed between instances in the new input QLF and instances in the existing model; finally, information consequent upon the input may be added, producing an updated discourse model (see Gaizauskas and Humphreys (1997) for further details).

Thus, in sum, the system takes a raw text as input and produces as output a meaning representation of the text which takes the form of a subgraph of a semantic net in which abstract representations of entities and events mentioned in the text are recorded, together with their attributes, and linked together via relations identified in the text or inferred from background conceptual/world knowledge modelled in the semantic net prior to the addition of the information from the text.

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 and then, in rank order, a predefined number of texts or passages retrieved for that question. When an answer is found, a response is written out and processing moves immediately to the next question, without considering alternate or perhaps superior answers that may lie in lower ranked documents. For the Inquery data, the top 10 passages[1] 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 TREC training set of questions (see next section). The system currently requires 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 yet been spent on optimisation.

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

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.

This question grammar distinguishes three basic syntactic question types:

1. subject *wh* questions consisting of a *wh* word or *whnp* (*wh* word acting as determiner plus noun phrase without determiner) followed by a verb phrase (e.g. *Who created the board game Pictionary? Which country has the largest part of the Amazon rain forest?*)
2. object *wh* questions consisting of a *wh* word or *whnp* followed by an inverted sentence (e.g *When did Nelson Mandela become President of South Africa?*)
3. non-*wh*-initial questions in which a *whnp* occurs as the complement of a verb or preposition in a non-initial position in the sentence (e.g. *The Faroes are a part of what northern European country?*)

The question grammar consists of 18 rules for these three questions types which were required to assign different semantics to different *wh* words and to deal with differences in active and passive forms. In addition we introduced three rules for inverted sentences and four rules for *whnp*s. The treatment was loosely inspired by Pereira and Shieber (1987) and 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.

[1]The passage width parameter in Inquery was set to 50 words and the best passage selected on this basis. A wider window of 250 words, with the 50 word passage at the centre, was then returned to QA-LaSIE to give it slightly more context.

Semantic features on each syntactic rule are used to build up a 'quasi-logical form' (QLF) representation compositionally during parsing, in the same way as for the rest of the grammar. Special semantic predicates, `qvar` (question variable) and `qattr` (question attribute), are used in the semantics to indicate the 'entity' about which the question seeks information and the attribute of that entity whose value is the information sought. For example, the question *Who composed Eugene Onegin?* would produce the following QLF representation:

```
qvar(e1), qattr(e1,name), person(e1)
person(e2), name(e2,'Eugene Onegin')
compose(e3), tense(e3,past), voice(e3,active),
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 `qvar(e1)` as well as `person(e1)` and `qattr(e1,name)`. 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.

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, the representation is processed as a special 'consequence' attribute to attempt to find an answer within the text's completed discourse model. Each question representation gives rise to a hypothesised entity (the `qvar`), and then the 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 be expected to have other attributes mentioned in, or inferred from the question, such as `name`, as well as 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 Gaizauskas and Humphreys (2000), 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 bringing all the problems of word sense ambiguity, a simple high-level general ontology was defined, and then reference made to WordNet (G.A. et al., 1993) 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.

Answer Generation An additional Q & A task-specific module was added to the LaSIE system, following the Discourse Interpreter stage. This module simply scans 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 longest realisation of the `qattr` attribute of that entity in the text is used as the central point from which 50- and 250-byte text windows were determined to be used as question responses (see next section); if no `qattr` is found the value of the `name` attribute is used as a default response. [2]

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. This approach suffers from two drawbacks:

1. The IR system's rankings may be faulty. Had QA-LaSIE processed all the top ranked texts or passages and found multiple answers, an independent mechanism for ranking the one or more answers returned could have been adopted, e.g., the similarity scores for `qvar` coreference.
2. It is 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.

TREC-8 Q & A Task and Evaluation Description

This section describes the TREC-8 Q & A task and gives details of the training and test setups, as well as the metrics used in the official evaluation and those we have introduced to help analyse our results more meaningfully.

The Task

The task is described quite simply in the task specifications (Que, 1999) as follows: "Given 200 questions, find their answers in a large text collection". The only further constraints were that each

[2] However, the longest realisation (in the case of multiple realisations resulting from coreference resolution) may not be the same mention of an entity as that which caused its selection as an answer. Thus, the answer string produced by QA-LaSIE may be centered around, say, 'International Business Machines', although the context in which the answer was found may only mention 'IBM'. It is not clear how such cases are judged if the answer string contains no context related to the question.

question is a single sentence, that the exact answer text occurs in at least one document in the text collection, and that the answer text is less than 50 bytes. All processing of questions as supplied by the track organisers and of the text collection was required to be fully automatic – no manual processing at any stage was allowed.

The data used for the evaluation were volumes 4 and 5 of the TREC Text Research Collection (see `http://trec.nist.gov`), excluding the Congressional Record. Thus, the data comprised approximately one gigabyte of news stories drawn from the *LA Times*, the *Foreign Broadcast Information Service*, the *Financial Times* and the *Federal Register* and covered a time period ranging from 1989 to 1996.

Two participation categories were defined: the 50 byte category and the 250 byte category. For each category systems could return up to 5 ranked answers. Each answer had to contain not just the answer text, but also a pointer (unique document number) to the document in the text collection in which the answer was found. In the 50 byte category an answer of at most 50 bytes was to be returned, and to be valid the string returned had to contain the entire answer. If the returned answer string contained more than one potential answer, then judgement as to the correctness of the answer was left to the human judges employed by NIST. In the 250 byte category an answer of a most one sentence or 250 bytes was allowed.

Training and Test Question Sets

Sample questions for the exercise were obtained from the participants several months before the evaluation by requiring each participating site to submit 10 questions, together with answers and pointers to the documents in which the answers were to be found. The track organisers then selected questions to be used for training and testing from these and from others introduced by themselves and the assessors, and also from the logs of the FAQFinder system (see Voorhees and Tice (1999) for full details of the question sources).

A training set of 38 questions was distributed to the participants some time before the final test. These questions fell into categories as follows: 8 questions demanding person names as answers (e.g. *Who was Johnny Mathis' high school track coach?*), 7 questions demanding dates as answers (e.g. *What year was the Magna Carta signed?*), 9 questions demanding lengths, heights, durations, or other measures/quantities as answers (e.g. *How tall is the Eiffel Tower?*), 8 questions demanding location names as answers (e.g. *What is the capital of Uganda?*), 5 questions demanding names of miscellaneous entities (companies, hotels, areas of the brain, species of pest, hurricanes), and one question requesting a shape description as answer (*What is the shape of a porpoises' tooth?*). For each of these training questions NIST supplied the document numbers of one or more texts in the text collection which contained an answer to the question.

The final test set consisted of 200 questions, 2 of which were excluded after the evaluation when judges found that no text in the data set contained an answer. We have not yet carried out a categorisation of the test set in order to keep the questions blind for use in evaluating future system development.

Metrics and Scoring

The correctness of each supplied answer was determined by a human judge at NIST. The judges were instructed to read each question and then for each candidate answer to decide initially whether it

was clearly correct or obviously wrong (without reference to the source documents). If a judge was uncertain as to whether a proposed answer was correct or incorrect then he or she was allowed to read the document from which the proposed answer was taken to see if, with that additional context, the answer proposed was credible.

The official metric for the task was the *mean reciprocal answer rank*, A, defined as

$$A = \frac{\sum_{i=1}^{N} r_i}{N}$$

where N is the number of questions and r_i is the reciprocal of the best (lowest) rank assigned by a system at which a correct answer is found for question i, or 0 if no correct answer was found.

Because our system returned at most one answer for each question and made the assumption (erroneously in light of the task definition) that some questions would have no answers in the document collection (and hence that systems could generate spurious answers for which they ought to be penalised), more sensible measures for it are the traditional measures of recall and precision which in this context may be defined as:

$$\text{Recall} = \frac{\text{number of correct answers}}{\text{number of questions to be answered}}$$

$$\text{Precision} = \frac{\text{number of correct answers}}{\text{number of questions answered}}$$

Note that the mean reciprocal rank metric reduces to recall, as we have defined it, in the case where only one answer per question is ever returned.

In the following section we adopt the recall and precision metrics; the system's "official" scores are identical to the recall figures.

Results and Analysis

In this section we present the results of QA-LaSIE in the test evaluation and in a more restricted evaluation we subsequently performed ourselves on the training data.

Test Questions

The following results were obtained from the individual judgements of question answers carried out by NIST and our own analysis of the system's output for each question.

For the NIST-supplied AT&T data, where the top 5 complete 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, 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 produced a correctly judged answer, though clearly there may be other retrieved texts which also contain an answer. 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 correct 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 only where texts containing an answer were correctly retrieved and where questions were parsed as interrogative constructions (i.e. where the question grammar produced a QLF representation of the question which included a `qvar`), which amounted to 122 of the 200 original questions. 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 above. 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%
```

Training Questions

In this section we present the results of the system on the 38 training questions. The system was exactly as used for the test evaluation, but did have the advantage that the question grammar was

developed by analysing a subset of the training questions. For this experiment judgements of correctness were made not by the NIST judges but by a member of our research group not involved in the development of the system. We considered only texts retrieved by the Inquery retrieval engine, as no data from the AT&T retrieval engine was available for the training questions.

Again using the top 10 passages for each question, the overall results on the 38 questions in the training set were:

```
50-byte answers:                    250-byte answers:
Recall    = 4 / 38 = 10.53%         Recall    = 8 / 38 = 21.05%
Precision = 4 / 18 = 22.22%         Precision = 8 / 18 = 44.44%
```

Analysing only those questions for which texts were correctly retrieved (here meaning that the IR results included a text either listed by NIST as including an answer, or in which a correct answer was found by the system) produced the following figures:

```
50-byte answers:                    250-byte answers:
Recall    = 4 / 24 = 16.67%         Recall    = 8 / 24 = 33.33%
Precision = 4 / 15 = 26.67%         Precision = 8 / 15 = 53.33%
```

The system parsed all but two of the training questions as questions (i.e. the question grammar produced a QLF representation which included a `qvar`). An answer was produced for one of these questions, via the fallback mechanism in the discourse interpreter, but it was judged incorrect.

Discussion

Failure Analysis

Only limited detailed analysis of system behaviour has been carried out to date, and we cannot yet make quantitative claims about where the strengths and weaknesses of the approach lie. We have investigated cases where the system succeeded on the training data, and in most of these cases it appears that the Inquery system was returning highly relevant texts and the QA-LaSIE system was performing semantic analysis of the question and these texts sufficiently well so as to identify correctly the semantic type of query variable (date, person, measure, etc.) and most entities of that type in the text. The degree of match that was required between query variable and instantiating entity in the answer text, i.e. the `qvar` coreference, was relatively undemanding. Thus, while in many cases constraints on the query variable beyond semantic type were not satisfied by the text, since the text was highly relevant and there were only a limited number of entities of the correct semantic type in the text, there was a reasonable probability that the one found and allowed by the weak matching process would be correct. Of course in other cases this overly liberal approach to matching the query variable against the text leads to spurious answers.

Beyond these initial observations, what is now required is a full programme for addressing sources of error in the system. Sources of error include:

- *Question Parsing* As only approximately two-thirds of the test set questions were parsed, more effort is needed to refine and extend the coverage of the question grammar. Further work is

also needed to refine the QLF semantics assigned to the question during grammatical analysis. For example, satisfactory semantic treatment of measure questions, pertaining either to objects or events (e.g. *How tall is the Eiffel Tower?*, *How long does it take to travel from London to Paris via the Chunnel?*), was not achieved. This needs to be addressed in conjunction with the appropriate treatment of semantics of measure expressions as found in answer texts (e.g. *300 meters tall/high, 300 meters in height, the iron-framed, 985ft 11in Eiffel Tower*, etc.).

- *QVAR Coreference* Analysis of whether `qvar` matching via 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 is too strong a requirement; on the other hand loosening the match results in spurious answers.
- *Answer Text Processing* Analysis needs to be carried out to see to what extent the discourse models ('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 syntactic or semantic analysis, inadequate lexical or world knowledge).
- *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.
- *Document/Passage Retrieval* The Inquery top 10 passage retrieval achieved slightly higher recall than the AT&T top 5 documents, but even so these top passages only contained answers for 76% of the questions. Since both of these systems made available considerably more texts (top 200 passages/100 texts respectively), an obvious exercise is to examine how many more answers are found in the residue and how far down the ranking they occur.
- *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, according to some measure independent of the IR systems' rankings. A single best answer or a ranked set of answers could then be returned. The impact of this on performance – both on precision and recall measured with respect to the highest ranked answer and on the mean reciprocal rank measure with respect to the top five answers per question – needs to be assessed.

At present we feel that the first two issues will be the most significant in future system development. Most other aspects of the system have been extensively tested in other applications and evaluations, such as MUC, but the QA-LaSIE system was assembled in less than two person weeks, and, in particular, very little effort was available to adapt the general coreference mechanism to the task of question resolution. We believe that considerable performance improvements on the Q & A task could be gained within the current approach.

Related Work

Given that the Q & A task ran for the first time at TREC-8 in 1999, there is not much work directly related to this task outside the TREC-8 participants' efforts. Forty-five runs were submitted to this

evaluation, though many of the 21 participating sites submitted multiple runs with essentially the same system. The scores on the evaluation ranged from a mean reciprocal rank of .002 to .660. The best QA-LaSIE run scored .111, placing the system in position 36 of the 45 runs evaluated. While this is by no means a satisfying result, given the limited effort put into the QA-LaSIE implementation we do not believe the evaluation results should be interpreted directly as an evaluation of the approach. Since QA-LaSIE returned only at most one result per question and scores for first-ranked answers for other systems are not available, direct comparison is not possible.

Reviewing all of the alternative approaches adopted in the TREC-8 Q & A task is clearly not possible here. We mention just the two highest performing systems, LASSO (Moldovan et al., 1999) and Textract (Srihari and Li, 1999). Both of these systems make use of shallow NLP techniques in conjunction with fairly conventional IR techniques – that is, a text retrieval engine is used to retrieve a subset of the document collection that is thought to hold the answers and then this subset is subjected to more intensive analysis in an attempt to extract answers. This is not unlike our overall approach, though in the case of LASSO considerable effort has gone into attempting to narrow down the texts examined for answers by expanding the query prior to submitting it to the search engine (no figures are available to indicate how successful this first, text retrieval stage was). Both systems also make use of shallow parsing techniques, both to analyse the question and the retrieved candidate answer texts/passages. In particular these systems make use of named entity analysers. However, while we attempt to match `qvar`s in the predicate-argument representation of a question against the predicate-argument representation of the text using a general coreference mechanism which does constraint-based matching, these approaches do a heuristic search around the most likely key words looking for expressions in the text which the shallow parser has identified as being of the correct semantic type. In essence these approaches are not deeply dissimilar to our own, but it appears LASSO and Textract put more effort into developing heuristics for the matching process, whereas we put more effort into trying to extract faithfully meaning representations from the questions and the answer texts, but did not refine the matching process. Clearly the former approach brings higher rewards, at least in the short term and given the nature of the Q & A task specification.

The TREC Q & A Task Revisited

While the Q & A task as conducted in TREC-8 was tremendously stimulating, there are several aspects of it that are open to question.

First, treating question answering as a task for which ranked answers are appropriate is debatable. Unlike "relevance" which is a notion which permits of degrees, correctness or incorrectness of answers does not. So, if a user asks a question such as *Who composed Eugene Onegin?* and gets back the ranked sequence of 5 answers *(Mussorgsky, Borodin, Tchaikovsky, Rimsky-Korsakov, Stravinsky)* this is not a great deal of use – he or she needs to go and check the related documents to find out the answer. Clearly narrowing a search to 5 documents is better than several thousands, but it is still contrary to the spirit of question answering which is to try to overcome the need to search retrieved documents for the answers to queries – otherwise why not just revert to document retrieval? As it stands the task is perhaps better described as "micro-passage retrieval" than question answering. Even if the utility of five ranked answers for development purposes is admitted it would be revealing to have as a separate metric scores for single (best) answers per question.

Secondly, the test set should contain at least some questions for which there are no answers in the test corpus. A system's ability to tell a user there is no answer to a question in a corpus is arguably as

important its ability to find one when it is there. If the QA track fosters the development of systems which assume that the document collections they address will always contain answers then this is liable to lead to the development of guessing behaviour which is inappropriate for real world scenarios. Adding questions for which there are no answers also allows for the introduction of a "precision" metric, since spurious answers will become possible (in the current setup systems are not penalised for guessing, so a not unreasonable baseline strategy would be to take the top document retrieved by a search engine when given the question as query and return the names of 5 entities found in it).

Finally, while some of the TREC-8 questions were "real" in that they were obtained from undergraduate students with a genuine interest in the answers, many were artificially created for the Q & A exercise and have the unfortunate characteristic of being back-formulations from the answer text to a question to which the text provides an answer. Thus it is to be hoped that future exercises will be able to find further sources of genuine questions.

Conclusion

We have described an approach to question answering that is based on linking an IR system with an NLP system that performs reasonably thorough linguistic analysis. Overall success is limited, but given the difficulty of the task, and the limited amount of development effort to date, this is not surprising. Thus, while the performance of the system leaves much to be desired, we believe that the approach performed sufficiently well and holds sufficient promise to warrant further investigation. Pursuing the lines of investigation discussed in our analysis of system failure in the previous section will help to reveal whether the approach we have followed for the Q & A task is appropriate. More generally it will shed light on the very interesting question this task poses: 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. We would also like to thank Mark Hepple and Michael Oakes of our department, the former for supporting the integration of WordNet into QA-LaSIE, the latter for making correctness judgements in the training question set evaluation. Finally, thanks to Mark Sanderson of Information Studies, University of Sheffield, for helping to interface the Inquery system with QA-LaSIE.

References

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.

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.

A. Copestake and K. Sparck Jones. Natural language interfaces to databases. *Knowledge Engineering Review*, 5(4):225–249, 1990.

Defense Advanced Research Projects Agency. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann, 1995.

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

R. Gaizauskas and K. Humphreys. Using a semantic network for information extraction. *Journal of Natural Language Engineering*, 3(2/3):147–169, 1997.

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*. John Benjamins, 2000.

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.

C. Green. Theorem proving by resolution as a basis for question-answering systems. *Machine Intelligence*, 4:183–205, 1969.

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.

D. Moldovan, S. Harabagiu, M. Paşca, R. Mihalcea, R. Goodrum, R. Gîrji, and V. Rus. LASSO: A tool for surfing the answer net. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 1999.

F.C.N Pereira and S.M. Shieber. *Prolog and Natural-Language Analysis*. Number 10 in CLSI Lecture Notes. Stanford University, Stanford, CA, 1987.

Question answering track at TREC-8. http://www.research.att.com/~singhal/qa-track-spec.txt, 1999. Site last visited November 6, 1999.

L.K. Schubert and L. Watanabe. What's in an answer: A theoretical perspective on deductive question answering. In *Proceedings of the Sixth Canadian Conference on AI*, pages 71–77, 1986.

A. Smeaton. Using NLP or NLP resources for information retrieval tasks. In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pages 99–111. Kluwer, Dordrecht, 1999.

K. Sparck Jones. What is the role of NLP in text retrieval? In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pages 1–24. Kluwer, Dordrecht, 1999.

R. Srihari and W. Li. Question answering supported by information extraction. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 1999.

T.Strzalkowski, F. Lin, J. Wang, and J. Perez-Carballo. Evaluating natural language processing techniques in information retrieval. In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pages 113–145. Kluwer, Dordrecht, 1999.

E. Voorhees and D. Tice. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 1999.