The York QA Prototype Question Answering System

Marco De Boni, Jose-Luis Jara-Valencia, Suresh Manandhar

Department of Computer Science,
University of York,
York Y010 5DD

DRAFT VERSION – DUE FOR REVISION

Abstract

A preliminary analysis of our QA system implemented for TREC-11 is presented, with an initial evaluation.

Introduction

The aim of our system in this year's TREC QA track was 1) to see how much our previous system could be improved simply by removing bugs and inconsistencies and 2) to test new techniques on "real" data. The system produced was therefore again a prototype experimental system rather than a "complete" machine ready for deployment. In particular it still lacks an information retrieval engine (we relied again on the documents retrieved by NIST's PRISE IR engine), and has only an outline Named Entity Recogniser and an incomplete answer extraction module. Nevertheless it did give an indication as to which ideas are most promising and should be investigated further, in particular as regards determining an answer in a sentence.

Improvements on the previous version

A close examination of the system used for TREC 2001 [Alfonseca et al. 2002] revealed a number of small bugs across all modules which were significant enough to affect performance. These were corrected for this year's entry. Furthermore, a close analysis of our results taking into account the contribution of each module revealed that a number of components we used did not improve performance and in fact were detrimental. In particular, the Noun Phrase Chunker which we used last year was removed as its output was not precise enough to be used productively.

The question recogniser which we previously used proved very hard to maintain and improve, based as it was on a large number of patterns and exceptions with very limited use of linguistic resources. It was therefore re-written from scratch in a much more elegant way in order to make much more use of linguistic resources such as WordNet. The Named Entity recogniser was also rewritten from scratch, as was the answer extractor, which had to cope with this year's track aim, which was to extract an exact answer as opposed to a string of words.

Question Recogniser

A simple set of rules allowed the question recogniser to determine the focus of the question. The initial recognition, however, was too refined to be of much use: given a question such as "What president's wife commented on the affair?", it would return a question type of "president's wife". The WordNet is-a hierarchy was therefore used to determine a less fine-grained answer type. WordNet, however, is deceptive without accurate word-sense disambiguation: satellite is a person, if satellite is intended in the sense of synset 107546753, "a person who follows or serves another", but not if it is intended in the more common meaning of synset 103275905, "a man-made object that orbits around the earth". Given that we did not have an accurate word-sense disambiguation module, we resorted to assuming that the meaning of any word was the most common word and hand-crafted a series of rules which reflected this (e.g. our rules state that "satellite is not a person, but is a thing").

On the other hand WordNet could not be used to determine whether a question was looking for MUC entities such as locations due to inconsistencies such as the fact that according to the hypernym hierarchy a city is a location, but a lake is not; again hand-crafted rules were applied to state facts such as "sea is a location" and hence determine that a question asking "What sea…" could be answered by looking for a location Named Entity.

Named Entity Recogniser

Texts were initially tokenized by applying several hand-coded heuristics. Then they were tagged using the TnT part-of-speech tagger (Brants 2000). Sentences were then split by an algorithm that utilises a number of heuristics and a list of abbreviations extracted by another algorithm that uses active learning when the case is highly ambiguous. There is no formal evaluation of this sentence splitter, but it appears to work well in most of cases.

For the Named entity recognition YorkQA system uses our own implementation of Nymble (Bikel et. al. 1997) which utilises hidden Markov models to identify named entities in the text. An improved version of this algorithm showed high accuracy in MUC-7. We estimate that this version of Nymble is reaching about 70% recall and 80% precision.

These results are a big improvement form YorkQA-2001. However, the program identifies only MUC entities (person names, organisation names, location names, dates, time expressions, currency expression and percentage expression). This means that for question about other types of entities (speeds, distances, durations, etc.) our system has much less information to work with.

We also detected that in question asking for fine-grain pieces of information (e.g. the first name of person), this module turned out to be inappropriate.

Semantic Distance Metrics

The semantic distance metric used for the TREC-10 was improved (see De Boni and Manandhar 2002 for an in-depth explanation of the implementation) and then subjected to a thorough analysis of its components. In particular we evaluated different approaches to the measure of semantic relevance for question answering in order to decide how the use of WordNet information, part-of-speech information and head-phrase chunking could provide a reliable measurement for semantic relevance. It was seen that a semantic distance metric based on all WordNet relations (is-a, satellite, similar, pertains, meronym, entails, etc) and using compound word information, proper noun identification, part-of-speech tagging and word frequency information gave best results.

Answer Word Finder

Our Named Entity recogniser only recognised a small number of entity types, which did not correspond to all types identified by the question type recogniser. In order to determine possible answers, the system made use of Wordnet's is-a relationships, looking in the answer sentence which had the highest semantic similarity measure for hypernyms of the question type. This approach appeared to be fruitful in determining "rare" entities which would be impractical to build a Named Entitiy recogniser to tag; a good example was the question asking the name of King Arthur's sword: the question type was "sword", the answer found was Excalibur as Excalibur was listed in WordNet as a hypernym of sword.

Conclusions and Further Work

This year's task confirmed that Question Answering is a complex task which requires accurate component parts: underperformance in any of the components results in a deterimental effect to the system as a whole. In such a complex system, however, it is not a simple task to determine why things go wrong and indeed why things go right. Nevertheless is is important to determine how the individual parts interact in order to improve performance, and, while such an analysis can be extremely laborious, it is necessary. The disadvantages of working in a very small team were also highlighted, as the complexity of the problem requires extensive work in disparate area.

Future work will include an extension an improvement of our Named Entity component, the use of an information retrieval engine (as opposed to relying on the documents provided by NIST), improvements in the answer locator and possibly some inference mechanism.

Bibliography

Alfonseca, E., De Boni, M., Jara, J.L., Manandhar, S., "A prototype Question Answering system using syntactic and semantic information for answer retrieval", in Proceedings of the 10th Text Retrieval Conference (TREC-10), Gaithersburg, US, 2002.

Bikel, D., Miller, S, Schwartz, R, and Weischedel, R. "Nymble: a high-performance learning name-finder". In Proceedings of the Fifth Conference on Applied Natural Language Processing, 194--201, 1997.

Brants, T., "TnT - A statistical Part-Of-Speech tagger", User Manual, 2000.

De Boni, M., and Manandhar, S., "Automated discovery of telic relations for WordNet", in Proceedings of the first International WordNet conference, India, 2002.