A semantic approach to Question Answering systems

Vicedo, Jose Luis & Ferrández, Antonio.

{vicedo,antonio}@dlsi.ua.es

Dpto. Lenguajes y Sistemas Informáticos. Universidad de Alicante

Campus de San Vicente del Raspeig

Apartado 99. 03080 Alicante, Spain

Abstract

This paper describes the architecture, operation and results obtained with the Question Answering prototype developed in the Department of Language Processing and Information Systems at the University of Alicante. Our approach accomplishes question representation by combining keywords with a semantic representation of expected answer characteristics. Answer string ranking is performed by computing similarity between this representation and document sentences.

1 Introduction

The prototype presented in this paper tries to face up question answering task from a new point of view. Question analysis obtains a mixed representation of queries based on keywords and a semantic representation of main information characteristics required by the question. Sentence ranking algorithm combines both representations to rank and select the best five answers. In the following section, our system is described. Afterwards, we analyse results obtained in TREC-9 Question Answering task. Initial conclusions are extracted and finally, directions for future work are discussed.

2 System Overview

Our system is structured into two main modules: *Question analysis module* and *Answer selection module*. First module processes questions expressed in open-domain natural language in order to obtain a representation of the information requested. This analysis is accomplished by obtaining *question type* and classifying terms into *keywords* and *definition terms*. Keywords help the system to locate sentences where answers can probably be found. A term in a query is considered a definition term if it defines characteristics of the expected answer. Question type and definition terms define the main information required by each question. A WordNet-based tool process questions type and definition terms in order to obtain a semantic representation of expected answer characteristics. This representation defines what we call *semantic context* of the target answer. The answer selection module uses keywords and semantic context to locate the sentence containing the answer and extract the part of the sentence that contains it. Figure 1 shows system architecture.

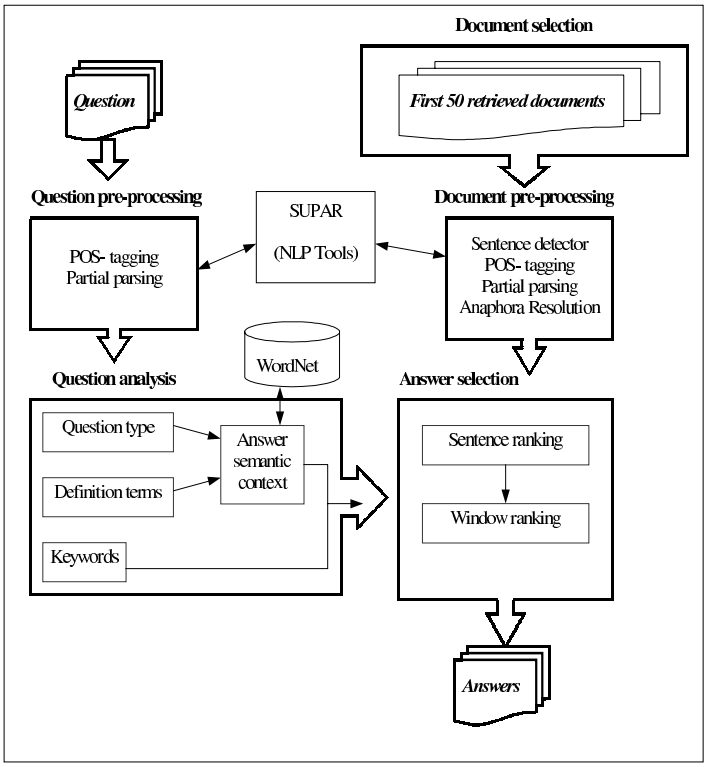

Figure 1. System architecture

2.1 Document Selection

We only processed the first fifty ranked documents supplied by TREC Organisation. Nevertheless, all QA track collection was analysed to obtain term *idf* weights (Salton, 1989). Term normalisation was performed using a version of Porter´s stemmer.

2.2 Question and Document pre-processing

Several Natural Language Processing techniques have been applied to both questions and documents. These tools compose the Slot Unification Parser for Anaphora Resolution (SUPAR) described in Ferrández (1999, 1998). SUPAR's architecture consists of three independent modules that interact with one other. These modules are lexical analysis, syntactic analysis, and a resolution module for NLP problems such as anaphora resolution. Queries and documents are pre-processed before entering question analysis and answer selection modules. Queries pre-processing consists on part-of-speech-tagging terms and parsing. Documents are pre-processed by detecting sentence boundaries, part-of-speech-tagging terms, parsing and solving pronominal anaphora. Managing with pronominal anaphora consists on substituting non-pleonastic third-person pronouns for their antecedents.

2.3 Question Analysis

Question processing module accomplishes different tasks. This module extracts main keywords, expands keyword terms, determines question type and builds the semantic context representation of the expected answer.

Question type is detected by analysing Wh-terms. This process maps Wh-terms into one or several of the following categories:

PERSON	GROUP	LOCATION	TIME
QUANTITY	REASON	MANNER	NONE

Each of these categories is related to WordNet top concepts (Miller, 1995). When no category can be detected by Wh-term analysis, NONE is used (e.g. "What" questions). This analysis gives the system three kinds of information: (1) lexical restrictions that expected answer should validate, (2) how to detect *definition terms* (if they exist), and (3) top WordNet concepts relevant to the expected answer.

Definition terms do not help the system to locate the correct answer but instead, they usually describe the kind of information requested. Depending on question type, different approaches are used to detect definition terms. For "What", "Which", "How", and similar questions these terms are detected by selecting noun phrases appearing next to the Wh-term. When questions such as "Find the number of…" or "Name a flying …" are analysed, noun phrases following the verb are considered definition terms.

Once question type and definition terms are analysed, the system generates the *semantic context of the expected answer*. A WordNet-based tool processes each definition term in order to build its semantic context representation. This context is represented as a weighted term vector that is computed as follows: for each definition, synonyms, one-level search hyponyms and all hyperonyms (until a top concept is achieved) are obtained. The weight assigned to these new terms is the *idf* of the analysed definition term in the collection divided by the distance in the WordNet hierarchy from this term to each new obtained one. When question type has been successfully mapped to a top concept, only terms related to this concept will be added to the term context representation. This way we obtain the terms that made up the context of a unique definition term. The semantic context representation of the answer (the joined representation of all definition term contexts) is computed by adding the context vector of each definition term in the question.

The semantic context of the answer helps the system in different ways: First, it approximates the type of the expected answer when the Wh-term analysis has been unable to obtain it. Second, as top concepts are too broad, it allows sub-classifying them for each particular question. And third, it helps the system to decide between different possible answers by comparing expected answer and probable answer semantic contexts.

To finish with question analysis, remaining question terms are considered keywords. When there are no remaining terms left (e.g. for the question "Name a flying mammal"), definition terms are used as keywords too. Non proper noun keywords are expanded using WordNet by adding to the question, keyword synonyms, one-level search hyponyms and one-level search hyperonyms.

2.4 Answer Selection

The input to this module is a small number of pre-processed candidate documents and the results of question analysis module. As first step, sentences are ranked accordingly to the following score:

*Sentence-score = Keyword_idf_sum + (0.65 * Expanded_keyword_idf_sum)*

where :

Keyword_idf_sum: is the sum of the *idf* weights for query keywords that appear in sentences.
Expanded_keyword_idf_sum: is the sum of the *idf* weights for terms obtained when expanding query keywords that also appear in sentences.

In both cases, the *idf* of a term that occur twice or more times in a sentence is added only once.

When this initial sentence ranking has finished, the first 100 ranked sentences that include probable answers are selected as the best candidates to contain the correct one. A term is considered a probable answer if it verifies lexical restrictions obtained by Wh-term analysis.

The final step is to analyse sentences to extract and rank the windows of the desired length that probably contain the correct answer. The system selects a window for each probable answer by taking as centre the term considered a probable answer. Each window is assigned a *window-score* that is computed as follows:

*Window-score = Sentence-score * (1+cos(Question_SC,Window_SC))*

where :

cos: Cosine
Question_SC: vector representing the semantic context of the expected answer.
Window_SC: vector representing the semantic context of terms contained into the selected window (excluding keywords and expanded keyword terms).

Finally, windows are ranked on window-score and the system returns the first five ranked windows as final result.

3 Results

TREC-9 Question Answering Track allowed five answers for each question. Besides, depending on answer-string length, two different run types were defined: up to 50 or 250 bytes long. We participated with two runs for each different answer length. Figure 2 shows results obtained. ALI9C runs have been produced applying the whole system described above. ALIC9A files contain results obtained applying the same strategy but without solving pronominal anaphora in relevant documents. These results were computed after getting rid of eleven questions whose answer did not appear in the document collection. Therefore, only 682 questions were evaluated.

Run	Answer lenght	Mean reciprocal rank strict	Mean reciprocal rank lenient	% Answers found strict	% Answers found lenient
ALI9C250	250	35,6%	37,1%	52,9%	55,3%
ALI9A250	250	34,9%	36,3%	51,6%	53,8%
ALI9C50	50	23,0%	24,5%	33,9%	36,1%
ALI9A50	50	22,7%	24,0%	33,9%	35,8%

Figure 2. QA Track results

Although a detailed results analysis is a very complex task, several conclusions can be extracted.

Retrieving relevant documents.
Correct answer was not included into the top fifty ranked documents supplied by TREC for 95 questions. As this fact relies on document retrieval strategy, we can not measure how our approach managed with these queries. Figure 3 analyses the percentage of questions that could be correctly answered depending on the number of top documents selected for searching the answer. Even if first 1000 documents were analysed, it would have been impossible to obtain the correct answer for 25 questions. It seems that document retrieval techniques do not fit QA retrieval needs. In fact, systems applying paragraph-indexing techniques (Harabagiu, 2000) (Clarke 2000) have obtained a better performance.

Top Docs Selected	10	25	50	100	250	500	750	1.000
Answer included	499	547	587	607	629	641	650	657
Answer Not included	183	135	95	75	53	41	32	25
% Answer Included	73,2%	80,2%	86,1%	89,0%	92,2%	94,0%	95,3%	96,3%

Figure 3. Document retrieval analysis

Context based answer detection.
Our main objective was to inspect how Wh and definition terms could be used to build a useful semantic representation of expected answer and if this representation could improve correct answer detection. Results analysis shows several circumstances. This approach increases system performance by comparing expected answer with probable answer contexts. Very good results are obtained when possible answers context gives some indication about the nature of these answers. In this case context analysis allows the system to find the correct answer, even to successfully decide between similar but different possible answers. However, when possible answer context does not include characteristics that define the possible answer, the system does not take profit of expected answer context definition. It seems clear that semantic context representation can not substitute the use of a Name–Entity tagger (not applied in our prototype). We think that combining both tools will contribute to improve system performance in two important aspects: (1) increasing the amount and quality of the information obtained from the question and (2) improving possible answers detection.

Another circumstance to take into account is the way of selecting terms that define the context of the possible answers. Nowadays, the system builds the semantic context of a possible answer from all terms included into the window (250 or 50 bytes) surrounding each probable answer. As results

show, results have become poorer as answer length decreased. This fact relies on the number and type of terms selected for building possible answer semantic context.

Pronominal anaphora resolution
Application of pronominal anaphora resolution has produced only a small benefit (around a 1%). Analysing this fact is very difficult but it relies on two main reasons. First, we have noticed that the number of relevant sentences involving pronouns is very low. And second, there are a lot of documents related to the same information, and sentences in a document that contain the right answer referenced by a pronoun, can also appear in another document without pronominal anaphora. Anyway, although the benefit is low, it can be considered a blind evaluation of how automatic pronominal anaphora resolution always helps QA systems performance.

4 Future Work

Several areas of future work have appeared while analysing results. First, IR system used for retrieving relevant documents has to be adapted for QA tasks. The IR used by TREC Organisation retrieved the document containing the correct answer into the first fifty relevant documents only for a 86% of the evaluated questions. Second, question analysis has to be improved by increasing the number of question types analysed (i.e. definition or list questions). Third, unless context based answer detection has revealed to help system performance it needs a finer tuning on defining the number and type of terms used for semantic context building and exploring the possibilities of a Name-Entity tagger. This strategy needs to be investigated and tested.

5 References

Clarke C., Cormack, G., Kisman D. and Lynam T. (2000) *Question Answering by passage selection*. In Proceedings of the Nineth Text Retrieval Conference. Washington (USA).

Ferrández A., Palomar M. and Moreno L. (1998) *Anaphora resolution in unrestricted texts with partial parsing*. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING – ACL. Montreal (Canada).

Ferrández A., Palomar M. and Moreno L. (1999) *An empirical approach to Spanish anaphora resolution*. To appear in Machine Translation.

Harabagiu S., Moldovan, D., Pasça M., Mihalcea R., Surdeanu M., Bunescu R., Gîrju R., Rus V., and Morărescu P. (2000) *FALCON: Boosting Knowledge for Answer Engines*. In Proceedings of the Nineth Text Retrieval Conference. Washington (USA).

Miller G.(1995), *"Wordnet: A Lexical Database for English"*, Communications of the ACM 38(11) pp 39-41.

Salton G.(1989), Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison Wesley Publishing, New York.

TREC-9, 2000. Call for participation Text Retrieval Conference 2000 (TREC-9). http://trec.nist.gov/cfp.html.