University of Alicante Experiments at TREC-2002[1]

Vicedo, Jose Luis & Llopis, Fernando & Ferrández, Antonio

{vicedo,llopis,antonio}@dlsi.ua.es

Dpto. Lenguajes y Sistemas Informáticos

Universidad de Alicante

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. This system is based on our TREC-10 approach where different improvements have been introduced. Main modifications reside on the introduction of a filtering stage into paragraph selection and answer extraction modules that allow the treatment of questions with no answer in the document collection. Moreover, WordNet has been enhanced by adding a collection of gazetteers that includes several types of proper nouns (people, organisations, and places) and a large variety of acronyms, measure and money units.

1. Introduction

This year, question-answering task has been significantly modified. The organisation has restricted the proposed experiments to *main* and *list* tasks. Main task is similar to previous year task but instead of permitting 5 ranked responses for each query and a maximum of 50 bytes as answer length, only a response is allowed and the answer string must contain nothing other than the exact answer. Besides, there is no guarantee that an answer will actually appear in the document collection. The list task consists of answering questions that will specify a number of instances to be retrieved. In this case, it is guaranteed that the collection contains at least as many instances as the question asks for.

The system presented to TREC-2002 QA task departs from the system presented in past TREC conferences [7][8] where new tools have been added and existing ones have been updated and adapted to cope with new specifications. Main enhancements rely on several aspects. First, passage selection and answer extraction stages have been adapted in order to face questions with no answer in the document collection. For this purpose, these stages have been complemented with a filtering module that rejects relevant paragraphs as well as possible answers that do not validate a series of restrictions. This way, when no possible answer remains after applying these restrictions, the system returns NIL as final answer. Second, WordNet has been extended by adding entities included in several gazetteers mainly referring to places (countries, states, cities, etc.) as well as and a large number of different acronyms, measure and money units. In this case, WordNet enrichment tries to minimise, as possible, the lack of a Name-Entity tagger.

Although our participation has been restricted to main task, this year we tried to face up all the specifications. In fact, it is the first time we manage with no-answer questions.

[1] This work has been partially supported by the Spanish Government (CICYT) with grant TIC2000-0664-C02-02 and (PROFIT) with grant FIT-150500-2002-416.

This paper is structured as follows: Section 2 describes system structure and operation and tries to emphasize new contributions. Afterwards, we present and analyse the results achieved and finally, we extract initial conclusions and discuss directions for future work.

2. System Overview

Our QA system is structured into four main modules: *question analysis*, *document/passage retrieval*, *paragraph selection* and *answer extraction*. First module processes questions expressed in open-domain natural language in order to analyse the information requested in the queries. This information is used as input by remaining modules. Document retrieval module accomplishes a first selection of relevant passages by using a passage retrieval system. Afterwards, the paragraph selection module filters these passages in order to select smaller text fragments (paragraphs) that are more likely to contain the correct answer. Finally, the answer selection module processes these fragments in order to locate and extract the final answer. Figure 1 shows system architecture.

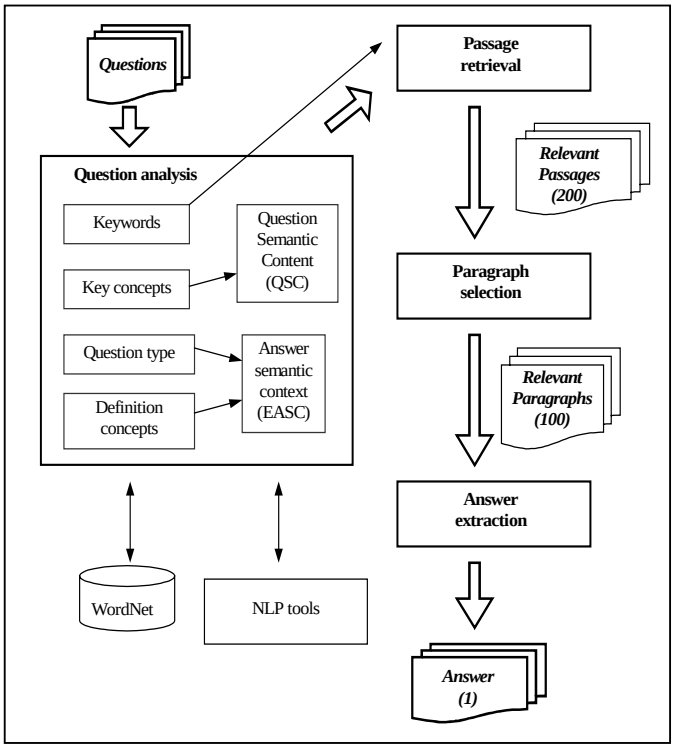

Figure 1. System architecture

2.1. Question Analysis

Question processing module accomplishes several tasks. First, questions are part-of-speech tagged and parsed. This process allows identifying simple noun and verbal phrases (*concepts*) in the query. Afterwards, this module determines *question type* and classifies concepts into two categories: *key* or *definition* concepts. Finally, these concepts are processed to obtain and represent its semantic characteristics. The resulting structures will be used as main information units for QA purposes.

Question analysis process starts with question type detection. This process maps Wh-terms (What, Which, How, etc) into one or several of the categories listed in figure 2. When no category can be detected by Wh-term analysis, NONE is used (e.g. "What" questions). This module also includes *definition* as a question type. Definition questions are detected by applying pattern-matching techniques.

Group A :	PERSON	LOCATION	GROUP	TIME	QUANTITY	NONE
Group B :	REASON	MANNER	DEFINITION			

Figure 2. Question type categories

Question type categories pertaining to *group A* are related to WordNet top concepts [2]. Each of these categories is represented by the vector of WordNet synsets that are semantically related to its corresponding top concept (called *QTC-Question type characteristics*). These synsets are obtained by extracting from WordNet all hyponyms of each top concept until third level and they are weighted depending on its level into the WordNet hierarchy and the frequency of its appearance into the path. As it can be deduced, NONE questions have an empty QTC.

Once question type has been obtained, the system selects the noun phrase in the query that expresses the semantic characteristics of the expected answer (*definition concept).* Definition concepts do not help the system to locate the correct answer into the document collection but they usually add critical information about the kind of information requested by the query. The semantic characteristics of definition concepts are represented by a weighted vector (*CT- concept type*) that includes the set of synsets they are semantically related to. These synsets are obtained by extracting from WordNet all hyperonyms of each definition concept head term (its path to top concepts) and they are weighted depending on its level into the WordNet hierarchy and the frequency of its appearance into the path towards top concepts.

QTC and definition concept CT are used to generate the *expected answer semantic context* (EASC). This context defines the semantic context that the expected answer has to be compatible with. This context is computed by performing exclusive vectorial addition between QTC and CT. As special case, EASC will be equal to CT for NONE questions. By the other hand, *group B* questions have a different treatment due to the special nature of its expected answers and therefore, at this analysis point only question type assignment is needed.

Once definition concepts have been detected, remaining question concepts are classified as *key concepts*. This question processing stage builds the semantic representation of the *key concepts* expressed into the query (*semantic content of a question - QSC*). This process consists of obtaining a general semantic representation of the concepts that appear in the question.

The head of a *key concept* syntactic structure represents the basic element or idea the concept refers to. Remaining terms pertaining to this structure modify this basic concept by refining the meaning represented by its head. Following this approach, the system tries to obtain and represent the different ways of expressing a concept. This process starts by associating each term pertaining to a concept, with its synonyms and one level search hyponyms and hyperonyms. These relations are extracted from WordNet lexical database. We define the semantic content of a term *t* (SC*t*) as a set of terms made up by the term *t* and all the terms related with it through the synonym and one level search hyponym and hyperonym relations. The SC of a term is represented using a weighted term vector. The weight assigned to each term pertaining to the SC of a term *t* is the 80%, 50% and

50% of the *idf* [3] value of term *t* for synonyms, hyponyms and hyperonyms respectively. As a concept is made up by the terms included into the same syntactic structure, we define the semantic content of a concept (SCC) as the set of weighted vectors (HSC, MSC) were HSC is the a vector obtained by adding the SC of the terms that made up the head of the concept and MSC is the vector resulting from adding the SC of terms that modify that head into the same syntactic structure. The set of SCCs that stand for the concepts appearing in a question builds the semantic content of a question (QSC). This way, the QSC represent all the concepts referenced into the question and the different ways of expressing each of them. All the described processes and related formulae are widely described and explained in [6].

Figure 3 sums up these processes using an example question. First, the system identifies and classifies the concepts *"company"*, *"manufacture"* and "*American Girl doll collection*" by parsing question. Afterwards, the system generates the expected answer semantic context (EASC) and obtains the semantic content of each key concept to compound semantic content of the question (QSC).

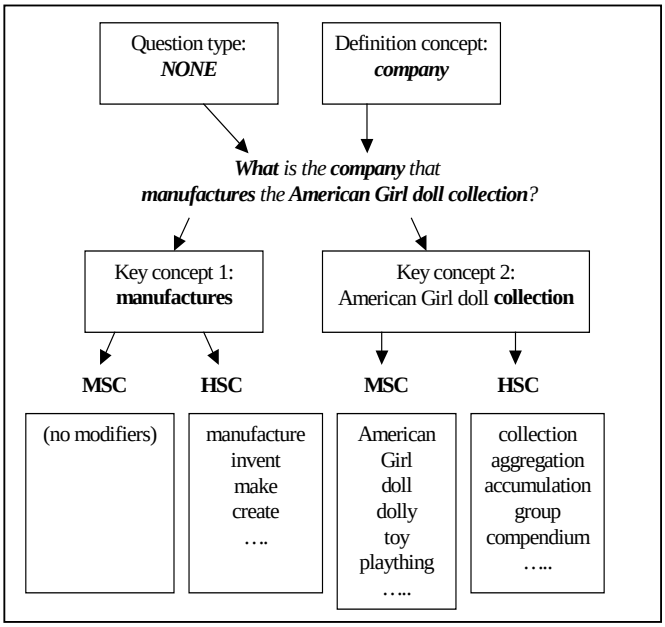

Figure 3. Question analysis processes

Question keywords are used for first stage passage retrieval, QSC information will help paragraph selection module to detect the paragraphs that are more likely to contain the answer and finally, EASC (or question type for group B questions) will allow detecting and evaluating possible answers.

2.2. Passage retrieval module

First stage retrieval applies the passage retrieval approach described in [1]. This passage retrieval can be applied over all the document collection, but it has only been applied for the 1,000 relevant documents supplied by TREC organisation. Therefore, keywords detected at question processing stage are used for retrieving the 200 most relevant passages from the documents included in this initial list. This process is intended to reduce the amount of text that has to be

processed by NLP modules since these passages are made up by text snippets of 15 sentences length.

2.3. Paragraph selection

This module processes the 200 first ranked passages selected at passage retrieval stage in order to extract smaller text fragments that are more likely to contain the answer to the query. As all this process is widely described in [7][9] we extract here the basic algorithm:

1. Documents are split into sentences.
2. Overlapping paragraphs of three sentences length are obtained.
3. Each paragraph is scored. This value measures the similarity between each paragraph and the question.
4. Paragraphs are ranked according to this score.

The score assigned to each paragraph (*paragraph-score*) is computed as follows:

a) Each SCC appearing in the question is compared with all the syntactic structures of the same type (noun or verbal phrases) appearing into each relevant paragraph. Each comparison generates a value. As result, each SCC is scored with the maximum value obtained for all the comparisons accomplished through the paragraph.
b) The paragraph-score assigned to each paragraph is obtained by adding the values obtained for all SCCs of the question as defined in previous step.
c) The value that measures similarity between a SCC and a syntactic structure of the same type is obtained by adding the weights of terms appearing into SCC vectors and the syntactic structure that is being analysed. If the head of this syntactic structure does not appear into the vector representing the SCC head (HSC), this value will be 0 (even if there are matching terms into MSC vector).

2.3.1. Filtering relevant paragraphs

This module has been added with the intention of managing with questions with no answer. This module filters relevant paragraphs by getting rid of those that contain value 0 for more than one SCC evaluated previously. This way, the system only accepts, as relevant, a paragraph that contains nearly all key concepts expressed in the query. At this stage, best 100 ranked paragraphs are selected to continue with the remaining processes.

2.4. Answer extraction

This process consists on analysing selected paragraphs in order to detect and evaluate concepts that can be considered probable answers. Among all the candidates the system will select the one it considers the correct answer. Answer extraction processes differ depending question type group.

For *group A* questions, the system gets rid of key concepts appearing in the paragraph and selects the concepts that validate lexical restrictions of the expected question (e.g. proper noun for a Who question). All these concepts are considered probable answers of the query. Next, the system computes the semantic context of each possible answer (SCPA) by taking into account the semantic concept types (CT) of the probable answer and its adjacent concepts in the paragraph. This way, The SCPA of a probable answer *r* is computed as:

$$SCPA_r = CT_{(r-1)} + CT_r + CT_{(r+1)}$$

Then, probable answers are filtered and only those that are compatible with the expected answer semantic context (EASC) are selected. For this purpose, each probable answer is assigned a score (*probable-answer-compatibility*) that measures its compatibility with the expected answer semantic context. Only probable answers with score greater than 0 are maintained. This value is computed as follows:

$$probable\text{-}answer\text{-}compatibility_r = cos(EASC, SCPA_r)$$

Next, compatible probable answers are evaluated by computing a final score (*answer-score*) that is obtained as follows:

$$answer\text{-}score_r = paragraph\text{-}score \cdot probable\text{-}answer\text{-}compatibility_r$$

Intuitively, the *answer-score* combines (1) the semantic compatibility between the probable answer and the expected answer (p*robable-answer-compatibility*) and (2) the degree of similarity between question and paragraphs (*paragraph-score*). Finally, probable answers are ranked on *answer-score* and the system returns the first one as correct answer or NIL when no compatible probable answers have been found.

Answer extraction manages differently with *group B* questions (*definition*, *reason* and *manner*). The answer to this kind of questions is usually a part of a sentence that defines a concept, reason or a way of performing an action and they are usually expressed via certain sentence syntactic structures. Consequently, our approach performs probable answer detection and extraction by applying syntactic pattern-matching techniques over relevant paragraphs. This way, when no pattern has been successfully validated, the system returns NIL as answer. This approach, as well as a full description of the patterns is described in [6].

3. Results

We submitted one single run for main task. This task allowed one answer for each question and the response had to contain only the exact answer string to be considered correct. Figure 4 shows the results obtained.

Number wrong:	302	
Number unsupported:	2	
Number inexact:	15	
Number right:	181	
Confidence-weighted score:	0.496	
Precision of recognizing no answer:		39 / 250 = 0.156
Recall of recognizing no answer:		39 / 46 = 0.848

Figure 4. TREC-2002 results

Our main objective was to inspect the way that restrictions imposed at paragraph selection and answer extraction stages affected system performance as a whole and, particularly, to the treatment of no-answer questions.

Result analysis shows two main circumstances to take into account. First, system performance presents a good precision since it has answered correctly a 72.4% of the questions the system considered to have answer in the collection (181 from 250 not NIL answers). Nevertheless, these filters seem to be too restrictive since the system has provided a NIL response for 211 questions with known answer. Second, our filtering approach does not perform correctly the detection of no answer questions. In fact, the precision achieved in this task has been very low (only a 15.6%). Moreover, despite of having answered as NIL a large number of questions (250), seven real NIL questions have not been recognised (a 15.2% of the 46 existing NIL questions).

Comparison with TREC-9 and TREC-10 results.

Comparison between our different participations is difficult and has to be analysed carefully since task specifications are significantly different. Nevertheless, we can compare 50-bytes strict results achieved in previous conferences with TREC-2002 results if we focus our attention mainly on the percentage of correct answers ranked in first place (see third column in figure 5). From this point of view, our system has achieved a significant improvement in precision since the percentage of correct answers retrieved in first place increases 12,8 points from TREC-10 results.

	% Answers found	% Answers in 1st place
TREC-9	33,9%	16,9%
TREC-10	39,6%	23,4%
TREC-2002	36,2%	36,2%

Figure 5. TREC Participation results

4. Future Work

As it can be deduced from result analysis, the main objective pursued this year has not been achieved. The filtering processes incorporated to paragraph selection and answer extraction stages have significantly increased system precision, they have failed in detecting questions with no answer in the collection. Consequently, we need to direct our next steps to investigate and test validation techniques that could cope efficiently with no answer questions.

5. References

1. Llopis F. and Vicedo J.L. *IR-n: a passage retrieval system at CLEF-2001*. In proceedings of the second Cross-Language Evaluation Forum (CLEF2001). Lecture Notes in Computer Science. September 2001. Darmstadt (Germany).
2. Miller G.(1995), *"Wordnet: A Lexical Database for English"*, Communications of the ACM 38(11) pp 39-41.
3. Salton G.(1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*

Information by Computer. Addison Wesley Publishing, New York.
4. TREC-9, 2000. Call for participation Text Retrieval Conference 2000 (TREC-9).
5. TREC-10, 2001. Call for participation Text Retrieval Conference 2001 (TREC-10).
6. Vicedo J.L. *SEMQA: Un modelo semántico aplicado a los sistemas de Búsqueda de Respuestas*. Phd Thesis. May 2002.
7. Vicedo J.L., Ferrandez A. And Llopis F. *University of Alicante at TREC-10*. In Proceedings of the Tenth Text Retrieval Conference. November 2001. Gaithersburg (USA).
8. Vicedo J.L. and Ferrandez A. *A semantic approach to Question Answering systems*. In Proceedings of the Ninth Text Retrieval Conference. November 2000. Gaithersburg (USA).
9. Vicedo J.L. *Using semantics for Paragraph selection in Question Answering systems*. In proceedings of the Proceedings of the Eighth String Processing and Information Retrieval Conference (SPIRE'2001). November 2001. Laguna de San Rafael (Chile).