ITC-irst at TREC-2003: the DIOGENE QA system

Milen Kouylekov, Bernardo Magnini, Matteo Negri, and **Hristo Tanev**
ITC-Irst, Centro per la Ricerca Scientifica e Tecnologica
Via Sommarive, 38050 Povo (TN), Italy
{kouylekov, magnini,negri,tanev}@itc.it

Abstract

This paper describes a new version of the DIOGENE Question Answering (QA) system developed at ITC-Irst. The recent updates here presented are targeted to the participation to TREC-2003 and meet the specific requirements of this year's QA main task. In particular, extending the backbone already developed for our participation to the last two editions of the QA track, special attention was paid to deal with the principal novelty factors of the new challenge, namely the introduction of the so-called *definition* and *list* questions. Moreover, we experimented with a first attempt to integrate parsing as a deeper linguistic analysis technique to find similarities between the syntactic structure of the input questions and the retrieved text passages. The outcome of such experiments, as well as the variations of the system's architecture and the results achieved at TREC-2003 will be presented in the following sections.

1 Introduction

The new version of DIOGENE described in this paper results from recent improvements to the well-tested backbone built in the framework of our participation to the last two editions of the TREC QA main task (see Magnini et al., 2001 and Magnini et al., 2002a) and to the first edition of the multiple language QA track at CLEF 2003 (Negri et al., 2003). This year, due to the availability of a relatively stable and reliable version of the system, most of the work concentrated on handling the new question classes introduced to complicate the TREC QA main task, namely the *definition* and *list* questions. To this aim, a specific module for definition questions (*e.g.* "*Who is Aaron Copland?*", "*What are fractals?*") has been created, which relies both on a set of specific hand-crafted answer patterns, and on the evaluation of the answer candidates through Web-based statistical techniques. Furthermore, as for list questions (*e.g.* "*Which past and present NFL players have the last name of Johnson?*"), the system was tuned by considering as correct answers all the candidates ranked over an experimentally determined threshold by the statistical answer validation component already described in (Magnini et al., 2002a).

Besides the *ad hoc* improvements specifically targeted to the TREC-2003 QA main task, some preliminary experiments were also carried out with the long-term goal of integrating parsing as a core technique to improve system's performance. More specifically, such integration is intended to improve part of the DIOGENE's basic components that usually fail when dealing with particular kinds of questions. For instance, *Answer-Type Identification* will benefit from the capability of finding more precisely the head of the (sometimes rather complex) NPs that follow the WH-word, as in "*What Boston Red Sox **infielder** was given his father's first name, with the letters reversed?*", or "*What country singer's first **album** was titled "Storms of Life"?*". Moreover, the introduction of parsing in the QA loop is a crucial step to refine the whole *Answer Extraction* process. With regard to this issue, while in the last year's version of DIOGENE this process was carried out only by considering the presence in a paragraph of named entities matching the answer type category, in the new version of the system we tried to consider the syntactic similarity between the input questions and the retrieved text passages as a further clue for candidate answers' selection. Being the extraction of answer candidates a critical issue, and one of the weakest modules in last year's version of DIOGENE, our experiments on parsing were mainly focused on this direction. The expected result was not only the improvement of system's performance over the types of questions it was already capable to deal with, but also some improvement over types of questions that the previous version of the system could not handle at all. As an example, this is the case of the very frequent (and apparently simple) questions whose answer is not a named entity (such as "*What instrument did Louis Armstrong play?*", and "*What color hair did Thomas Jefferson have before gray?*") and the "*HOW-DID*" questions (such as "*How did Jimi Hendrix die?*"), which represent a challenging direction for future developments.

Starting from these general premises, this paper will mainly describe the novelties and the experiments carried out to develop this year's version of DIOGENE. In particular, after a short overview of the system's architecture (Section 2), we will focus on the new module developed to handle definition questions (Section 3), and on the experiments carried out to use parsing as a technique to refine the answer extraction process (Section 4). Finally, Section 5 will conclude the paper presenting the results achieved by DIOGENE at TREC-2003, as well as some final remarks about strengths and weaknesses of our system.

2 DIOGENE's Architecture

The overall system's architecture (depicted in Figure 1) relies on the rather standard three-components backbone used for the participation to the last two editions of the TREC QA main task. Such a backbone relies on a *question processing* component, which is in charge of the linguistic analysis of input questions, a *search* component, which performs the query composition and the document retrieval, and an *answer extraction* component, which extracts the final answer from the retrieved text passages (see (Magnini et al., 2002a) for further details).

Within this rather conservative framework, the automatic answer validation technique developed last year still plays a crucial role. The algorithm, fully described in (Magnini et al., 2002a), relies on discovering relations between a question and the answer candidates by mining the Web or a large text corpus for their co-occurrence tendency. Summarizing, the answer validation process is carried out through the following steps:

1. Compute the set of representative keywords *Kq* and *Ka* both from the question and the answer.
2. From the extracted keywords compute a set of *validation patterns* (*i.e.* textual expressions where the question and the answer keywords co-occur closely).
3. Submit the validation patterns to the Web.
4. Estimate an *answer relevance score (ARS)* considering the number of retrieved documents.

The *ARS* is calculated on the basis of the number of *hits* (*i.e.* retrieved pages) by means of a statistical co-occurrence metric called *corrected conditional probability* (Magnini et al., 2002b). The formula we used is the following:

$$ARS(a) = \frac{P(Ka \mid Kq)}{P(Ka)^{2/3}} = \frac{hits(pattern(Ka,Kq))}{hits(Kq) * hits(Ka)^{2/3}} * \left|EnglishPages\right|$$

Such a general formula had to be specified to deal with the new kinds of questions presented in this year's edition of the TREC QA main task. In particular, while *factoid* questions were handled with the original *ARS* calculation formula, *list* questions required the experimental definition of a relative threshold to select a larger number of answers, and *definition* questions required the development of *ad-hoc* validation patterns. While the experimental setting of a threshold to capture relevant answers to an input list question is a relatively easy task, let's focus on the more interesting and challenging issue of answering definition questions.

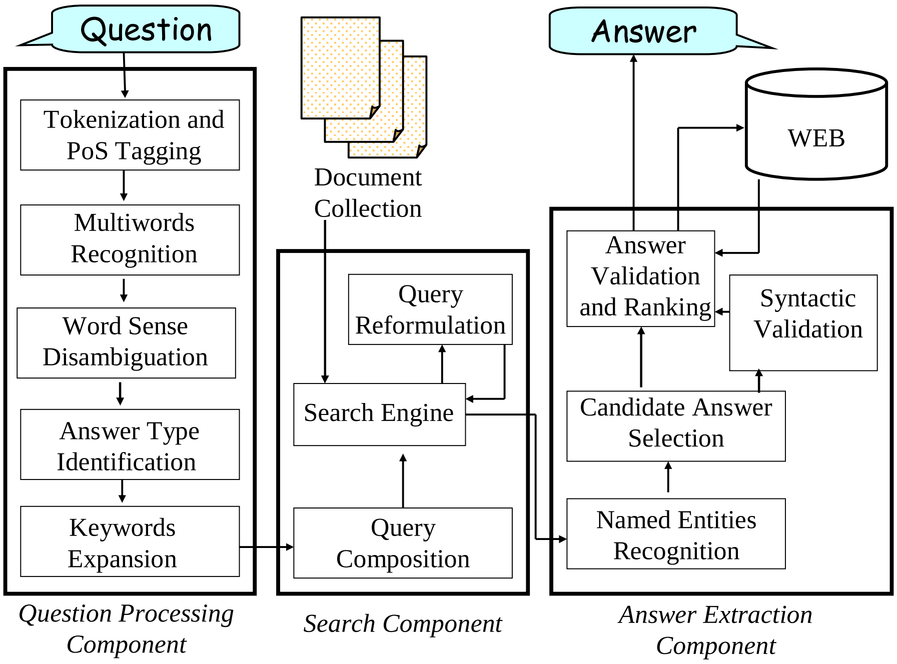
Question
Tokenization and PoS Tagging
Multiwords Recognition
Word Sense Disambiguation
Answer Type Identification
Keywords Expansion
Question Processing Component
Document Collection
Query Reformulation
Search Engine
Query Composition
Search Component
Answer
WEB
Answer Validation and Ranking
Syntactic Validation
Candidate Answer Selection
Named Entities Recognition
Answer Extraction Component

Figure 1. DIOGENE's Architecture.

3 Answering Definition Questions

In this year's TREC QA main task competition 10% of the questions belonged to the type definition. However, according to the evaluation scheme adopted, these questions contributed to the overall score up to 25%, thus forcing participants to invest on this particular aspect of research in QA. Our strategy relies on using patterns to extract the best text fragments where definitions are likely to occur (we call them "definition fragments") and then going to the Web to measure the co-occurrence between the question *focus* (*i.e.* the entity for which a definition is sought, such as "golden parachute" in the question "What is a golden parachute?") and the most important part of the definition (*i.e.* the so-called "definition core"), usually represented by an NP contained in the definition fragment.

3.1. Extraction and ranking of DEFINITION-FRAGMENTs

At the beginning of the process we use a small set of manually defined lexical patterns to extract and rank definition-like fragments. Being these patterns weighted, our technique resorts to calculating a score for every candidate fragment summing the weights from the matching.
For instance, the most used patterns for the extraction of candidate DEFINITION-FRAGMENTs are the following:

FOCUS {"who"|"what"|"which"} {"is"|"was"} DEFINITION-FRAGMENT
nonPrep FOCUS {"is"|"was"} DEFINITION-FRAGMENT

nonPrep FOCUS (DEFINITION-FRAGMENT)
nonPrep FOCUS, DEFINITION-FRAGMENT
DEFINITION-FRAGMENT “known as” FOCUS
DEFINITION-FRAGMENT “called” FOCUS

where “DEFINITION-FRAGMENT” stands for the part we take for further processing and “nonPrep” stands for any word which is not a preposition. We also used, as further clues, the presence of hypernyms of the focus and words from the WordNet gloss, in case the focus is found in WordNet hierarchy.

The following step consists of sorting all the DEFINITION-FRAGMENTs according to their score and passing them to the next module which is in charge of querying the Web.

3.2. Extraction and validation of definition cores (DEF-COREs)

At this stage of the process, we consider as possible appropriate answers to a definition question all the noun phrases that appear close to the question focus in the DEFINITION-FRAGMENTs retrieved from the document collection. For example, given the DEFINITION-FRAGMENT:

“… The Italian skier Alberto Tomba won the World Cup in 1993…”

the corresponding candidate answer phrases will be “Italian skier” and “World Cup”. We think that such noun phrases, to which we refer with the term DEF-CORE, represent the core of a good definition or at least an introduction to it. DEF-COREs are extracted from the candidate passages by means of the shallow parser Scoll (Abney, 1996).

Once the DEF-CORES have been extracted, their validation (*i.e.* the calculation of the corresponding *Answer Relevance Score*) is carried out automatically by means of the statistical answer validation technique outlined in Section 2. However, such technique allows that only one simple validation pattern, namely the generic pattern [Kq NEAR Ka], is considered. By definition, this pattern will lead to the number of pages where the keywords from the question (Kq) appear close to the keywords from the answer (Ka). However, for each specific question type it is possible to define one or more validation patterns which are much more efficient than the generic validation pattern. In particular, for the definition questions we can use the following list of more precise validation patterns (where Kq and Ka have been respectively substituted with FOCUS and DEF-CORE):

FOCUS “is” {“”|”a”|”an”|”the”} DEF-CORE
FOCUS “was” {“”|”a”|”an”|”the”} DEF-CORE
FOCUS “means” {“”|”a”|”an”|”the”} DEF-CORE
FOCUS “stands for” {“”|”a”|”an”|”the”} DEF-CORE
FOCUS “known as” {“”|”a”|”an”|”the”} DEF-CORE

These patterns are intended to present the typical lexical context used by an English speaker to introduce common notions when giving a definition for an entity. Moreover, even though some of them show a limited applicability with respect to some possible definition questions (*e.g.* patterns like “means” and “stands for” can not be applied to validate questions whose focus is a person name), all of them are completely domain independent. During the development we considered also other kinds of patterns, but we decided not to use them as they didn’t bring enough statistically relevant improvements to the performance.

In order a DEF-CORE to be taken into consideration, at least for one of the patterns the search engine should return relevant documents; this way, the number of pages where the focus and the DEF-COREs co-occurr is the combined number of the pages for all the patterns.

During the validation, the DEF-COREs are striped from determiners and are tested with all possible determiners. Often the DEF-COREs contain too many adjectives that make them receive

zero score when retrieving relevant documents. In this case we calculate the score for any of the sub-phrases of the DEF-CORE and take the maximum score obtained.
In light of these assumptions, the A*nswer Relevance Score (ARS)* measure is specified in the following way:

$$ARS(def-core)=\frac{P(def-core\mid focus)}{P(def-core)^{2/3}}=\frac{\sum_{patterns} hits(pattern(def-core,\, focus))}{hits(focus)*hits(def-core)^{2/3}}*\left|EnglishPages\right|$$

which gives a score to the candidate DEF-COREs through the statistic measure of their co-occurrence with the focus.

3.3. Discussion, comments and future work

The proposed algorithm for extracting answers to definition questions gave us rather promising results. Even though the overall score for the definition part of the question set was not very high, with an average F score of 0.317 for the fifty definition questions of the competition, we think that the evaluation scheme that we have presented gives an appropriate framework for answering such type of questions. Our analysis of the results shows that on 76% of the questions the system has provided a correct answer. The major problems came from the relatively low recall (only 38% with respect to the vital nuggets selected by the NIST assessors as ideal answers). This is probably due to the fact that the methodology that we presented is more oriented towards canonical definition, rather than important facts and events related to the question focus which was the main objective of the TREC organizers when creating the appropriate nuggets. This leads us to the general conclusion that we need a more linguistically oriented approach, more focused on deep analysis of candidate answers. Another problem of our approach is related to the velocity with which the search engine returns the relevant documents. However our opinion is that using a large source of information as the Web is important to extract good answers to definition questions.

4 Experiment: Adding Parsing in the QA Loop

This year, we integrated in DIOGENE's architecture an algorithm for graph matching between syntactic structures in order to add structural-semantic criteria to the answer validation process which up to now was entirely based on techniques exploiting the Web redundancy. For every candidate answer, the graph matching algorithm gives a score which reflects structural, lexical and semantic similarities between its syntactic context and the question. The main assumption behind the use of a parser for answer validation is that often the question and the syntactic context of the answer have similar structures. Resorting to this assumption, besides our short-term goal of improving the answer validation process, our experiments represent a preliminary step towards the long-term goal of dealing with questions whose answer is not a named entity.
Given two parse trees, the main scope of the graph matching algorithm here presented is to find the best mapping among the two, considering similarities among their lexical content.
In the following explanation we will present the graph matching algorithm using as an example the question-answer_passage pair:

Question #1920: “When was ‘Cold Mountain’ written?”
Answer passage: “When the ‘Cold Mountain’ began rising to the top of bestseller lists in 1997…”.

Our algorithm proceeds through the following steps:

0. The syntactic structures both of the question Q(V_Q,E_Q) (vertices V_Q and edges E_Q) and the candidate answer passage, CA(V_{CA},E_{CA}), are found. To this aim, we use dependency parse trees produced by the RASP (Carrol and Briscoe, 2002) parsing toolkit (in reality the structures are not trees but rather directed acyclic graphs).
1. An association graph is built A(V_A,E_A) with a set of vertices V_A and edges E_A. Such association graph generalizes the structure of both input graphs - Q and CA.

   1.1. Every vertex from the association graph A has two corresponding vertices from both graphs Q and CA. From any two vertices $v_Q \in V_Q$ and $v_{CA} \in V_{CA}$ we may form a vertex $v_A \in V_A$ if the lexical or part-of-speech tag labels on v_Q and v_{CA} are consistent and can be generalized. In such case the label on v_A is a generalization of the labels on v_Q and v_{CA}. For example, in Figure 2 the vertices from Q and CA labeled with "Cold Mountain" generate in the corresponding association graph a vertex with the same label. Moreover, being both of them verbs, the vertices "written" from Q and "rising" from CA generate a vertex labeled with "V" in A. In the model we have adopted, two words which have the same part of speech can also be generalized if they have a common hypernym in WordNet.

   1.2 We put an edge between any two vertices $v_A{}^1$ and $v_A{}^2$ from V_A if their corresponding vertices from Q and CA are linked in the same direction.
   Formally this means:

$$IF\ \ v_A^1 = generalize\ \ (v_Q^1, v_{CA}^1)\ \ AND\ \ v_A^2 = generalize\ \ (v_Q^2, v_{CA}^2)\ \ THEN$$

$$(v_A^1, v_A^2) \in E_A \Leftrightarrow (v_{CA}^1, v_{CA}^2) \in E_{CA}\ \ AND\ \ (v_Q^1, v_Q^2) \in E_{CA}$$

2. We have defined a function *weight* which gives a score to every syntactic structure obtained via generalization of two structures. In this way we can define for every substructure of A, a weight which is based on the number of edges and vertices and the labels they have. The best match is defined as the highest weighted sub-graph of A in which no vertices share common corresponding vertex in Q or CA. Considering the table in Figure 2, the last condition can be formulated in the following way: all the vertices in the matching sub-graph of A have to be distributed on different columns and rows. We call the resulting sub-graph the *best matching graph* of Q and CA.

3. From the question form we define the possible syntactic position of the exact answer (in Figure 2 the position is denoted with X). If the candidate exact answer matches the position X, then it takes the score of the matching between Q and CA.

4.1 Calculating the weights

As it was mentioned before, when two vertices are generalized in one vertex in the association graph, we assign a generalizing label to this *association vertex.* According to the differences in the labels of the vertices which have been generalized, we assign a score to the *association vertex*.

For every vertex in the association graph $v_A = generalize(v_Q, v_{CA})$ we calculate its weight by means of the following heuristically defined parameters:

$$weight(v_A) = \begin{cases} \bullet\ 300, \mathit{if\ the\ lexical\ part\ of}\ v_{CA}\ \mathit{and}\ v_Q\ \mathit{is\ equal\ and\ they\ both\ represent\ names} \\ \bullet\ 100, \mathit{if\ the\ lexical\ part\ of}\ v_{CA}\ \mathit{and}\ v_Q\ \mathit{is\ equal\ and\ it\ is\ not\ a\ name} \\ \bullet\ 100*k,\ \mathit{if}\ v_{CA}\ \mathit{and}\ v_Q\ \mathit{have\ a\ common\ hypernym\ in\ WordNet} \\ \mathit{in\ this\ case\ k\ is\ defined\ from\ the\ number\ of\ vertices\ between\ the\ common} \\ \mathit{hypernym\ and\ the\ generalized\ words} \\ \bullet\ 1.5, \mathit{if}\ v_{CA}\ \mathit{and}\ v_Q\ \mathit{have\ the\ same\ part\ of\ speech} \\ \bullet\ 0, \mathit{if\ none\ of\ the\ above\ conditions\ hold} \end{cases}$$

We define the weights of the edge e_A in the association graph as:

$$weight(e_A) = \begin{cases} \bullet\ 2, \mathit{if\ both\ association\ vertices\ it\ connects\ represent\ a\ lexical\ item\ which\ is\ not} \\ \mathit{a\ stop-word} \\ \bullet\ 1, \mathit{if\ only\ one\ of\ the\ vertices\ is\ a\ lexical\ item\ which\ is\ not\ a\ stop-word} \\ \bullet\ 0, \mathit{if\ none\ of\ the\ above\ conditions\ hold} \end{cases}$$

Using these definitions, we can define the weight of any syntactic sub-graph A'(V',E') of the association graph A(V,E) in the following way:

$$weight(A'(V',E')) = \left(\sum_{e \in E'} weight(e) \right)^2 . \sqrt{\sum_{v \in V'} weight(v)}$$

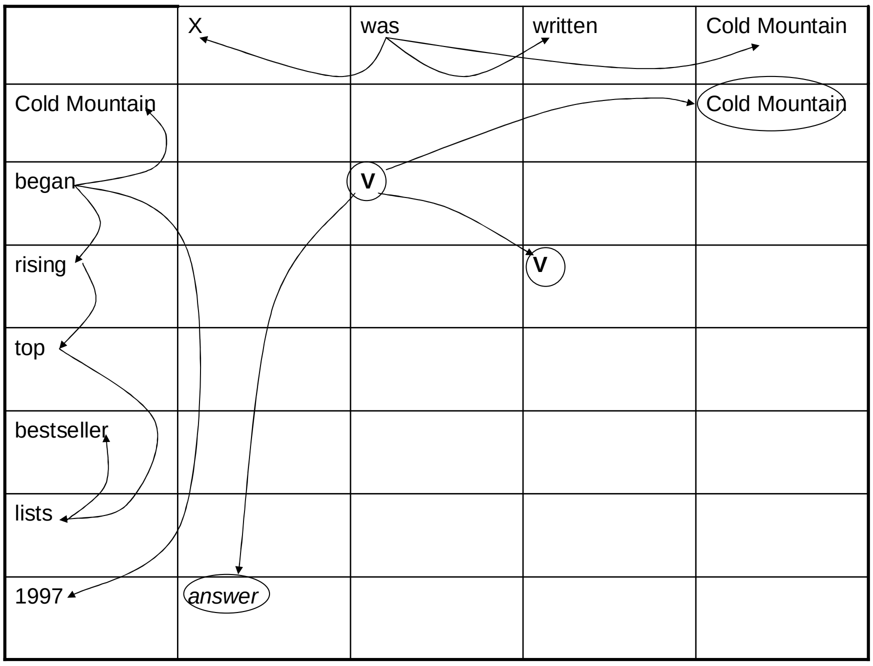

Figure 2 Matching among syntactic trees

Finally, we calculate the weight of every candidate answer passage by calculating the weight of the *best matching graph* between the sentence where it appears and the question. The algorithm gives this score only if the candidate answer matches the answer position in the question (denoted by X in Figure 2).

4.2 Problems and discussion on the syntactic graph matching

We did not have enough time to precisely define the parameters of the syntactic graph matching described so far; therefore, the application of the syntactic validation criteria gave no improvement in the overall result. Although, we did not perform complete error analysis, the following weak points of the current implementation can be pinpointed:

1. We are still not able to identify the position of the expected answer (X in Figure 2) with enough precision.
2. Calculation of the weights is not refined and parameters are only intuitively defined. For example, it would be much better to define the weight of the vertices considering their IDF value in a corpus.
3. Syntactic and lexical transformations can be integrated in the algorithm in order to make the matching of the structures more flexible (for instance, considering nominalization of verbs and active-passive transformations could improve our method).
4. Anaphora and ellipsis should be resolved before applying the syntactic structure matching; unfortunately, these techniques were not implemented in this version.
5. We did not normalize the question by translating it in affirmative form; this also influences the precision of the approach.

However, our empirical observations show that structural similarities between the question and the candidate answer passages often exist, and can be identified by inexact graph matching techniques. Therefore, fine tuning of the parameters of the matching algorithm will be necessary to identify these similarities and use them to improve both our answer ranking criteria and the overall answer extraction process.

5 Results and Conclusion

DIOGENE's performance has been evaluated over the three runs submitted to the TREC-2003 QA main task (see Table 1).

	Factoid							**List**	**Definition**	**FINAL SCORE**
	W	U	X	R	Accuracy	NIL Prec.	NIL Rec.	Average F	Average F	
irstqa2003w	300	10	6	97	0.235	0.121	0.267	0.076	0.317	**0.216**
irstqa2003p	305	11	5	92	0.223	0.132	0.167	0.074	0.315	**0.209**
irstqa2003d	343	4	4	62	0.150	0.111	0.067	0.067	0.318	**0.171**

Table 1: DIOGENE at TREC-2003

As for the 413 factoid questions, the total number of wrong (W), unsupported (U), inexact (X) and right (R) answers, together with the overall accuracy, the precision and the recall of recognizing NIL answers are reported for each run. Results for the list and the definition questions are only presented in terms of the average F-measure scores achieved over the total number of questions (respectively 37 and 50).

All these results have been obtained using the same overall architecture, but varying the validation technique to answer factoid and list questions. In particular:

irstqa2003w, our best run, has been obtained relying on the Web as the unique resource to accomplish automatic evaluation as in last year's best performing version of DIOGENE.

irstqa2003p results from the combination of the scores provided by the Web-based answer validation methodology and the graph-matching technique described in Section 4. Unfortunately, this did not bring any improvement to the system's performance. This is probably due to the weaknesses of the approach already mentioned in the same section. Nevertheless, in light of our empirical observations, parsing and other deeper linguistic analysis techniques (*e.g.* anaphora resolution, temporal and spatial reasoning, etc.) are deemed necessary to deal with the general QA problem and, more specifically, with the increasing difficulty level of the TREC competition.

irstqa2003d, surprisingly the worst result, has been obtained combining the Web-based answer validation technique with metrics that take into account the density of the query keywords within the retrieved passages. Our surprise, partially motivated by the higher difficulty of this year's TREC questions, comes from the fact that the same combined validation technique proved to be the most successful in the recent multiple-language QA track at CLEF-2003 (see Negri et al. 2003 for details).

A general conclusion that can be drawn in light of these results is that statistical approaches are relatively easy to implement and prove to be effective for some of the QA subtasks such as answer validation, allowing systems to reach reasonable performances with a limited effort. However, as they are limited to the statistically relevant knowledge that we can acquire from the Web or from an off-line corpus, deeper linguistic techniques seem to be a crucial step towards higher flexibility, coverage, and effectiveness of any QA system.

References

Abney, S.: Partial Parsing via Finite-State Cascades. Proceedings of the ESSLLI '96 Robust Parsing Workshop (1996)

Carroll, J., Briscoe, E.: High Precision Extraction of Grammatical Relations. Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002), Taipei, Taiwan (2002).

Magnini, B., Negri, M., Prevete, R., Tanev, H.: Multilingual Question/Answering: the DIOGENE System. Proceedings of the Tenth Text Retrieval Conference (TREC-10), Gaithersburg, MD. (2001).

Magnini, B., Negri, M., Prevete, R., Tanev, H.: Mining Knowledge from Repeated Co-occurrences: DIOGENE at TREC-2002. Proceedings of the Eleventh Text Retrieval Conference (TREC-2002), Gaithersburg, MD. (2002a).

Magnini, B., Negri, M., Prevete, R., Tanev, H.: Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002), Philadelphia, PA. (2002b).

Negri, M., Tanev, H., Magnini, B.: Bridging Languages for Question Answering: DIOGENE at CLEF-2003. Proceedings of the Cross Language Evaluation Forum (CLEF-2003), Trondheim, Norway (2003).