Mining Knowledge from Repeated Co-occurrences: DIOGENE at TREC-2002

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

Abstract

This paper presents a new version of the DIOGENE Question Answering (QA) system developed at ITC-Irst. With respect to our first participation to the TREC QA main task (TREC-2001), the system presents both improvements and extensions. On one hand, significant improvements rely on the substitution of basic components (*e.g.* the search engine and the tool in charge of the named entities recognition) with new modules that enhanced the overall system's performance. On the other hand, an effective extension of DIOGENE is represented by the introduction of a module for the automatic assessment of the candidate answers' quality. All the variations with respect to the first version of the system as well as the results obtained at the TREC-2002 QA main task are presented and discussed in the paper.

1 Introduction

The new version of the DIOGENE QA system described in this paper is based on last year's version (Magnini et al., 2001), focusing on two main directions: the improvement of its basic components and the extension of the original architecture.

First, the architecture of the system has been improved by substituting part of its modules and algorithms with more suitable and reliable solutions. Since an analysis of the information flow throughout the process indicated the *search component* and the *answer extraction component* were the main error sources in our previous participation to the competition, most of the improvements concern these aspects of the architecture. In particular, a new search engine, new document indexing techniques, new query formulation criteria and a new module for named entities recognition have been adopted.

Second, the system has been extended by adding a module for a fast and totally automatic evaluation of candidate answer strings (Magnini et al., 2002a). The main reason behind the necessity of providing the system with an answer validation component concerns the difficulty of picking up from a document the "exact answer" required by the TREC-2002 main task guidelines. Moreover, one of the lessons learned after our first participation to the TREC QA main task was the importance of a reliable distinction between possible correct answers and the huge quantity of spurious material retrieved by the search engine. As an example, given the question "*Who is Tom Cruise married to?*" and the text snippet "*Married actors Tom Cruise and Nicole Kidman play Dr. William and Alice Harford, a wealthy New York couple who think their eight-year marriage is very, very good.*", we had to deal with the difficulty of understanding who is the real Tom Cruise's wife and select the exact answer "Nicole Kidman" among the candidates. Our approach to automatic answer validation 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. The underlying hypothesis is that the number of these co-occurrences can be considered a significant clue to the validity of the answer. As a consequence, this information can be effectively used to rank the huge amount of candidate answers that our QA system is often required to deal with. Considering the above example, the introduction of the automatic answer validation component met the specific need of providing DIOGENE with an effective way of filtering out the improper candidate "Alice Harford" and choosing the best exact answer within the document retrieved by the search engine.

Since the overall system's architecture is slightly similar to the one described in (Magnini et al., 2001), this paper will be mainly focused on the description of the novelties of this year's version of DIOGENE.

After a short description of the *question processing component* in Section 2, the main features of our new *search component* will be presented in Section 3. Then, the *answer extraction component* will be thoroughly analyzed in Section 4, with a particular emphasis on the details of our approach to automatic answer validation (Section 4.2). Section 5 will conclude the paper illustrating the results achieved by our system at TREC-2002, providing a preliminary error analysis and some final remarks about strengths and weaknesses of DIOGENE.

2 Question Processing Component

Apart from the introduction of the answer validation component, the architecture of DIOGENE has not changed. The system still relies on three basic components (see Figure 1), namely the *question processing* component (in charge of the linguistic analysis of input questions), the *search component* (which performs the query composition and the document retrieval), and the *answer extraction* component (which extracts the final answer from the retrieved text passages).

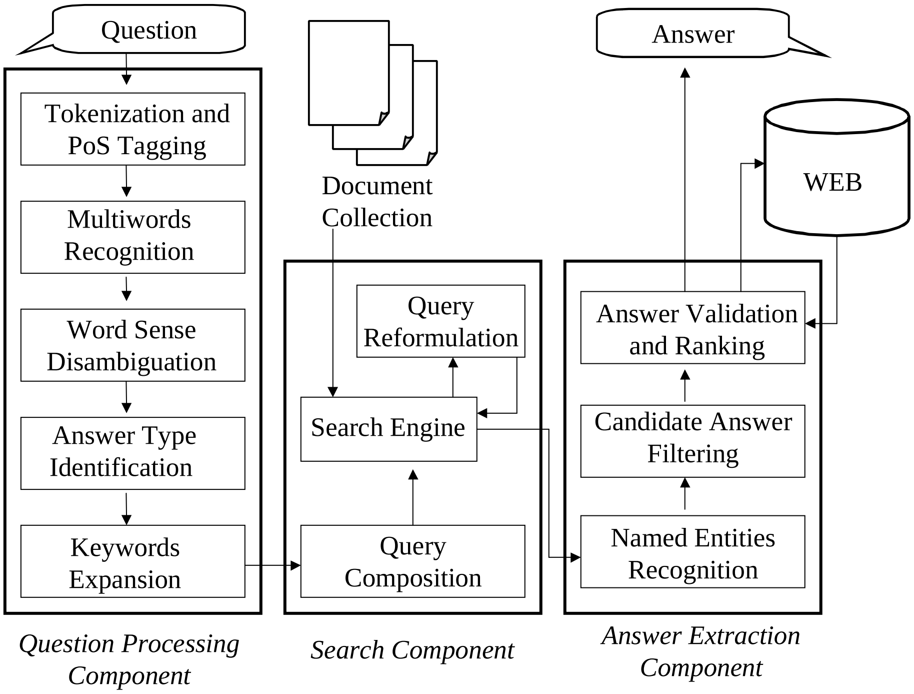
Question
Tokenization and PoS Tagging
Multiwords Recognition
Word Sense Disambiguation
Answer Type Identification
Keywords Expansion
Question Processing Component
Document Collection
Query Reformulation
Search Engine
Query Composition
Search Component
Answer
WEB
Answer Validation and Ranking
Candidate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

Figure 1. DIOGENE Architecture.

During the *question processing* phase, the linguistic analysis of the input is performed sequentially by the following modules:

- ***Tokenization and PoS tagging***. First, the question is tokenized and words are disambiguated with respect to their lexical category by means of the Treetagger (Schmid, 1994), a statistical Part of Speech tagger developed at the University of Stuttgart.
- ***Multiwords recognition***. About five thousand multiwords (i.e. collocations, compounds and complex terms) have been automatically extracted from WORDNET (Fellbaum, 1998) and are recognized by means of pattern matching rules.
- ***Word sense disambiguation***. The identification of the correct sense of the question terms is necessary to expand the search query with synonyms of that words without the risk of introducing

disturbing elements. In order to provide correct synonyms for a reasonable query expansion, question words are disambiguated with respect to their senses.

- ***Answer type identification***. The answer type for a question represents the entity to be searched as an answer: this information is used to select the correct answer to an input question within the documents retrieved by the search engine. In particular, knowing the category of the entity we are looking for (e.g. PERSON, LOCATION, DATE, etc.) we can determine if any "candidate answer" found in a document is an appropriate instantiation of that category. Our answer type identification module relies on a manually defined taxonomy of answer types (e.g. "PERSON", "LOCATION", "ORGANIZATION", "DATE", etc.) and a set of approximately 240 rules that check different features of the input question. Answer type identification is the aspect of the question processing component that presents the most significant improvements with respect to last year's version of DIOGENE. In order to enlarge the coverage of possible types of questions and refine the answer type taxonomy, more than 100 rules have been added to the original module used for our first participation inn the TREC competition.
- ***Keyword expansion***. At the end of the linguistic processing of the question, a stop words filter is applied that cuts off both non-content words and non-relevant content words. The remaining words (we call them "basic keywords") are then passed to an expansion phase which considers both morphological derivations and synonyms.

3 Search Component

The search component first composes the question keywords and their lexical expansions in a Boolean query, then performs document retrieval from the AQUAINT text collection. In order to overcome the main difficulties encountered last year using the Zprise search engine provided by NIST (i.e. the lack of Boolean expression support, as well as the necessity of considering a maximum of ten keywords per query and retrieving a maximum of 10 documents per question in order to bring the processing time under control), this year we adopted Managing Gigabytes (MG) (Witten et al., 1999) as a new search engine. MG is an open-source indexing and retrieval system for text, images, and textual images covered by a GNU public license and available via ftp from *http://www.cs.mu.oz.au/mg/*.
Besides the speed of the document retrieval, the advantages derived from using MG are twofold. First, it allows for the customization of the indexing procedure. As a consequence, we opted to index the AQUAINT collection at the paragraph level, using the paragraph markers provided in the SGML format of the documents. This way, although no proximity operator (e.g. the "NEAR" operator provided by AltaVista) is implemented in MG, the paragraph index makes the "AND" Boolean operator perform proximity search. In order to divide very long paragraphs into short passages, we set 20 text lines as the limit for paragraphs' length. This new indexing criterion allowed us to avoid the huge quantity of errors related to the paragraph filtering techniques used in last year's version of DIOGENE.
The other advantage derived from using MG concerns the possibility of performing Boolean queries, thus obtaining more control over the terms that must be present in the retrieved documents. Using the Boolean query mode, at the first step of the search phase all the basic keywords are connected in a complex "AND" clause, where the term variants (morphological derivations and synonyms) are combined in an "OR" clause. As an example, the question "*When did Titanic sink?*" is transformed into:

[**Titanic** AND (**sink** OR **sank** OR **sunk**)]

However, Boolean queries often tend to return too many or too few documents. To cope with this problem, we implemented a feedback loop which starts with a query containing all the basic keywords and gradually simplifies it by ignoring some of them. Several heuristics are used by the algorithm. For example, a word is removed if the resulting query does not produce more than a fixed number of hits (this probably means that the word is significant). Other heuristics consider the capitalization of the query terms, their part of speech, their position in the question, WordNet class, etc. (see Magnini et al., 2002b). The algorithm stops when a maximum of 150 text paragraphs has been collected or a certain percentage of the question terms has been cut off. This way, the searching algorithm builds a set of the most significant words and narrows it until enough documents are retrieved. The efficiency of these kinds of feedback loops has been recently pointed out by (Harabagiu et al., 2001).

4 Answer Extraction Component

The answer extraction component is the other aspect of the architecture that has been considerably updated. As stated before (Section 1), most of the efforts were dedicated to the substitution of the named entities recognition module (the performance analysis of the tool used in the last years' version of the system showed a 60% error rate), and the extension of DIOGENE with a module for the automatic evaluation and ranking of the answer candidates. Both of the new modules are described in the following sections.

4.1 Named Entities Recognition

Once the relevant paragraphs have been retrieved, the named entities recognition module is in charge of identifying within these text portions all the entities that match the answer type category (e.g. PERSON, ORGANIZATION, LOCATION, MEASURE, etc.). The task is performed by a rule-based named entities recognition system for the English written language developed at ITC-Irst (Magnini et al. 2002c). The core of the system relies on the combination of a set of language dependent rules with a set of predicates, defined on the WORDNET hierarchy for the identification of both proper names (i.e. person, location and organization names, such as "Galileo Galilei", "Rome", and "Bundesbank") and *trigger words* (i.e. predicates and constructions typically associated with named entities, such as "astronomer", "capital", and "bank").

The process of recognition and identification of the named entities present in a text is carried out in three phases. The first phase (*preprocessing*) performs tokenization, PoS-tagging, and multiwords recognition in the input text. In the second phase, a set of approximately 200 *basic rules* is used for finding and marking with SGML tags all the possible named entities present in the text (e.g. <MEASURE><CARDINAL>*200*<\CARDINAL> *miles*<\MEASURE> *from* <LOCATION>*New York*<\LOCATION>). Finally, a set of higher level *composition rules* is used to remove inclusions and overlaps among tags (e.g. <MEASURE>*200 miles*<\MEASURE> *from* <LOCATION>*New York*<\LOCATION>) as well as for co-reference resolution.

The system has been tested using the test corpora and the scoring software provided in the framework of the DARPA/NIST HUB4 evaluation exercise (Chinchor et al., 1998). Results achieved over a 365Kb test corpus of newswire texts vary among categories, ranging from an F-Measure score of 71% for the category MEASURE, to 96.5% for the category DATE.

4.2 Answer Validation

The answer validation module is in charge of evaluating and scoring a maximum of 40 answer candidates per question in order to find the exact answer required as the final output. The top 40 answer candidates are selected, among the named entities matching the answer type category, on the basis of their distance from the basic keywords and their frequency in the paragraphs retrieved by the search engine.

The basic idea behind our approach to answer validation is to identify semantic relations between concepts by mining for their tendency to co-occur in a large document collection. In this framework, considering the Web as the largest open domain text corpus containing information about almost all the different areas of the human knowledge, all the required information about the relation (if exists) between a question *q* and an answer *a* can be automatically acquired on the fly by exploiting Web data redundancy.

In particular, given a question *q* and an answer *a*, it is possible to combine them in a set of *validation statements* whose truthfulness is equivalent to the degree of relevance of *a* with respect to *q*. For instance, given the question "*What is the capital of the USA?*", the problem of validating the answer "*Washington*" is equivalent to estimating the truthfulness of the validation statement "*The capital of the USA is Washington*". Therefore, the answer validation task could be reformulated as a problem of statement reliability. There are two issues to be addressed in order to make this intuition effective. First, the idea of a validation statement is still insufficient to catch the richness of implicit knowledge that may connect an answer to a question. Our solution to this problem relies on the definition of the more flexible idea of a *validation pattern*, in which the question and answer keywords co-occur closely. Second, we need an effective and efficient way to check the reliability of a validation pattern. With regard to this issue, we propose two solutions relying on a statistical count of Web searches and on document content analysis respectively. A detailed description and a comparison between the experimental results achieved by the two approaches is presented in (Magnini et al., 2002a).
With reference to the above considerations, given a question-answer pair [*q*,*a*] we propose the following generic scheme for answer validation. Both the statistical and the content-based approach perform four basic steps:

1) Compute the set of representative keywords *Kq* and *Ka* both from *q* and from *a*. This step is carried out using linguistic techniques, such as answer type identification (from the question) and named entities recognition (from the answer);
2) From the extracted keywords construct the validation pattern for the pair [*q*,*a*];
3) Submit the validation pattern to a search engine;
4) Estimate an *Answer Relevance Score (ARS)* considering the results returned by the search engine.

The retrieval on the Web is delegated to a publically available search engine (e.g. AltaVista or Google). The post-processing of the results is performed by HTML parsing procedures and simple functions which calculate the *ARS* for every [*q, a*] pair by analyzing the results pages returned by the search engine. The two algorithms for automatic answer validation diverge in the methodology for the *ARS* calculation as well as for the search engine used; nevertheless, in both cases Web documents are not downloaded, thus making the algorithms rather efficient.

Statistical approach. The pure statistical approach makes use of the AltaVista search engine (http://www.altavista.com), exploiting the proximity operator "*NEAR*" to retrieve only Web documents where the answer and the question keywords co-occur. The *ARS* is then calculated on the basis of the number of 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:

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

where:

- *hits(Ka NEAR Kq)* is the number of English-language pages returned by AltaVista, where the answer keywords (*Ka*) and the question keywords (*Kq*) are in distance of no more than 10 words of each other;
- *hits(Kq)* and *hits(Ka)* are the number of English-language pages where *Kq* and *Ka* occur respectively;
- |*EnglishPages*| is the number of English pages, indexed by AltaVista.

This formula can be viewed as a modification of the Pointwise Mutual Information formula, a widely used measure that was first introduced for identifying lexical relationships (in this case the co-occurrence of *Kq* and *Ka*).

Content-based approach. The content-based approach makes use of Google (http://www.google.com), taking advantage of the text passages (i.e. snippets) returned by the search engine as output of a Web search. Using the fact that Google ranks higher the documents where the query terms co-occur close to each other, the *ARS* is calculated considering the presence of relevant keywords within the top 100 retrieved snippets. The underlying assumption is that the closer the distance between the candidate answer *a* and the question keywords within these text passages, the stronger their relation is.
Every appearance of the candidate answer *a* in a snippet is evaluated by calculating a *co-occurrence weight*, as the number of the question keywords and their distance from *a*. If we have co-occurrence of the answer *a* and a set of question keywords QK = $\{qk_1, qk_2, ...\}$ the co-occurrence weight CW(*a*,QK) is calculated by means of the following formula:

$$\mathrm{CW}(a,\mathrm{QK}) = \prod_i w(qk_i)^{(\| qk_i a \| + 1)^{-1}}$$

Where:

- $w(qk_i)$ is the weight of the question keyword qk_i. In general, $w(qk_i)$ can be calculated from the keyword frequency. However in the current implementation of the algorithm we used equal weights for all the words.
- $\| qk_i a \|$ denotes the distance between the answer *a* and the closest appearance of qk_i.

If we denote with S_a the set of the top 100 text snippets, the *ARS* is calculated through the formula:

$$ARS = \sum_{QK \in S_a} CW(a, QK)$$

This formula gives high preference to the answers which occur close the question keywords.

As stated before, validation patterns capture the relation (if one exists) between a question and an answer through simple co-occurrence mining. However, an additional pattern-based approach exploiting the information conveyed by the presence within the Web documents of explicit validation statements (e.g. phrase patterns such as "*The capital of the USA is Washington*") has been partially explored. In this version of DIOGENE, the use of a kind of phrase pattern slightly similar to the ones described in (Subbotin and Subbotin, 2001) has also been tested for the simplest possible variant of the "*Where is*" questions. In particular, if the question is of the type "*Where is <NP>?*" (where <NP> stands for a simple noun phrase without attached prepositional phrases), we search the Web for the phrase pattern ["<NP> in *a*"]. The number of hits produced by the search is then used to increase the *ARS*.
This solution proved to be rather effective and, in some cases, allowed DIOGENE to avoid errors produced by the simple co-occurrence mining techniques. As an example, given the question "*Where is the Orinoco River?*" and the two answer candidates "*Amazon*" and "*Venezuela*" both the statistical and the content-based approaches gave preference to "*Amazon*" as the best final answer. However, a Web search with the string ["Orinoco River in Amazon"] did not find any documents, while the string ["Orinoco River in Venezuela"] returned respectively 322 hits using Google and 104 using AltaVista, thus confirming that the location of the Orinoco River is Venezuela. In this case, the *ARS* obtained considering the presence of the phrase pattern ["Orinoco River in Venezuela"] into the Web documents led DIOGENE to the correct answer.
In general, the exploitation of different levels of patterns, ranging from the more general validation patterns to the more specific phrase patterns is of great interest, and seems to be a simple and powerful

instrument for answer extraction and validation. Exploitation of such patterns requires a very detailed question taxonomy and the development of machine learning techniques for their automatic acquisition.

4.3 Answer ranking

This year the ranking of the answers to the 500 questions of the QA main task was of great importance for the final score. In fact, a measure that is an analogue to the document retrieval's uninterpolated average precision was used to score the runs. The *Confidence-Weighted Score* (*CWS*) formula gives higher weights to the answers for which systems are more confident (i.e. answers with a higher rank in the run submissions), thus penalizing systems unable to accomplish a reliable calculation of the answers' confidence level.

As stated before, DIOGENE exploits the results of the answer validation also at the answer ranking phase. The task is accomplished combining the ARS with a *Question Type Reliability* (*QTR*) coefficient which indicates how reliable the ARS is with respect to a given question type. As an example, the QTR associated with the questions asking for a PERSON is 1, while the QTR for ORGANIZATION questions is 0.5 and for LOCATION questions is 0.75. The QTR coefficient for the different question types was computed considering the results of the answer validation experiments described in (Magnini et al 2002a). Given the QTR and the ARS, the confidence level (CFL) is calculated by the following formula:

if the answer is not NIL , $CFL = QTR * ARS$
if the answer is NIL , $CFL = QTR * 0.1$

where NIL means that no answer was found in the target corpus.

Since the ARS is usually much higher than 1, this formula gives a lower preference to the NIL answers, pushing them toward the end of the run submission (in our submission, the first NIL answer was ranked 406th). This is motivated by the fact that, as we process only a maximum of 150 paragraphs per question, there is no certainty about the correctness of the NIL answers. As a consequence, giving the NIL answers a lower rank reduces the impact of errors caused by the possible false negatives.

# Answers	Right	Unsupported	Inexact	Wrong
1 to 100	73	7	8	12
100 to 200	44	10	5	41
200 to 300	39	2	3	56
300 to 400	20	4	1	75
400 to 500	16	1	0	85
Total	192	24	17	267

Table 1. Distribution of Right, Unsupported, Inexact, and Wrong answers.

Table 1 shows how correct, unsupported, inexact and wrong answers have been ranked by DIOGENE in the best of the three runs submitted. Results confirm that our answer ranking technique performed well, producing an output list where most of the correct answers are distributed at the top (73% of the top ranked 100 answers are correct).

5 Results and Discussion

DIOGENE's performance has been evaluated over three runs submitted to the TREC-2002 QA main task (see Table 2). The three answer lists have been produced using the same architecture, simply by varying the answer validation algorithm in order to test the impact of the different approaches. The best classified run (with a confidence-weighted score of 0.589, around 6% above the other two) was obtained using the content-based answer validation approach, while the second classified resulted from the combination of the statistical and the content-based techniques, and the third resulted from the application of the statistical approach.

Run	Right	Unsupported	Inexact	Wrong	CWS
IRST02D1	192	24	17	267	0.589
IRST02D3	177	23	16	284	0.533
IRST02D2	173	19	14	294	0.520

Table 2. ITC-Irst at TREC-2002.

In order to evaluate strengths and weaknesses of DIOGENE, an error analysis was carried out considering the first 100 questions where the system failed. Also this year, most of the errors (40%) came from incorrect document retrieval. An in depth analysis of the search phase results revealed two main sources of errors. First, the stemming algorithm used by MG leads to the retrieval of many irrelevant paragraphs. Second, many errors are due to the difficulty of dealing with the variety of lexical formulations of an answer with respect to a question. The solution to this problem requires the development of intelligent query formulation criteria, going beyond the simple algorithms for keyword extraction and query expansion with synonyms and morphological derivations. For instance, reliable query formulation criteria should consider the relation between the question semantics and the possible transformations of its surface form. In spite of the remarkable improvements brought to the overall system's performance, answer validation is the reason for 38% of the errors. Most of these errors came from the fact that our approach measures the co-occurrence between the entities and does not consider the semantic relation which is the origin of that co-occurrence. As an example, given the question "*What is Buzz Aldrin's real first name*?", our answer validation component returned "*Neil Armstrong*" (the person name most frequently co-occurring with the question keywords) instead of the correct answer "*Edwin*". The answer candidates extraction produced 19% of the errors. In some cases this is due to the fact that DIOGENE is still unable to determine the correct answer type for some classes of questions (i.e. "Why" and "How" questions, such as "*Why does the moon turn orange*?", and "*How did Mahatma Gandhi die*?"), thus providing a huge number of irrelevant answer candidates. In some other cases the correct candidates have been discarded because of their distance from the query keywords within the retrieved paragraphs or because of errors in the named entity recognition phase. Also this year, the answer type extraction module performed well, with an error rate of only 3% due to PoS-tagging and disambiguation errors.

References

Chinchor, N., Robinson, P., Brown, E.: Hub-4 Named Entity Task Definition (version 4.8). Technical Report, SAIC. *http://www.nist.gov/speech/hub4_98.*

Fellbaum, C.: WORDNET, An Electronic Lexical Database. The MIT Press (1998).

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girjiu, R., Rus, V., Morarescu, P.: The Role of Lexico-Semantic Feedback in Open-Domain Question-Answering. Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001), Toulouse, France (2001).

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.: Comparing Statistical and Content-Based Techniques for Answer Validation on the Web. Proceedings of the VIII Convegno AI*IA, Siena, Italy, (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).

Magnini, B., Negri, M., Prevete, R., Tanev, H.: A WordNet-Based Approach to Named Entities Recognition. Proceedings of SemaNet02, COLING Workshop on Building and Using Semantic Networks, Taipei, Taiwan, (2002c).

Moldovan D., Harabagiu S., Pasca M., Girju R., Goodrum R., Rus V.: The Structure and Performance of an Open-Domain Question Answering System. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL), Hong Kong, (2000).

Schmid, H.: Probabilistic Part-Of-Speech Tagging Using Decision Trees. Proceedings of the International Conference on New Methods in Language Processing (1994).

Subbotin, M., Subbotin, S.: Patterns of Potential Answer Expressions as Clues to the Right Answers. Proceedings of the Tenth Text Retrieval Conference (TREC-10), Gaithersburg, MD. (2001).

Witten, I. H., Moffat, A., Bell T.: Managing Gigabytes: Compressing and Indexing Documents and Images (second ed.), Morgan Kaufmann Publishers, New York (1999)