Is It the Right Answer?
Exploiting Web Redundancy for Answer Validation

Bernardo Magnini, Matteo Negri, Roberto Prevete and **Hristo Tanev**
ITC-Irst, Centro per la Ricerca Scientifica e Tecnologica
[magnini,negri,prevete,tanev]@itc.it

Abstract

Answer Validation is an emerging topic in Question Answering, where open domain systems are often required to rank huge amounts of candidate answers. We present a novel approach to answer validation based on the intuition that the amount of implicit knowledge which connects an answer to a question can be quantitatively estimated by exploiting the redundancy of Web information. Experiments carried out on the TREC-2001 judged-answer collection show that the approach achieves a high level of performance (*i.e.* 81% success rate). The simplicity and the efficiency of this approach make it suitable to be used as a module in Question Answering systems.

1 Introduction

Open domain question-answering (QA) systems search for answers to a natural language question either on the Web or in a local document collection. Different techniques, varying from surface patterns (Subbotin and Subbotin, 2001) to deep semantic analysis (Zajac, 2001), are used to extract the text fragments containing candidate answers. Several systems apply answer validation techniques with the goal of filtering out improper candidates by checking how adequate a candidate answer is with respect to a given question. These approaches rely on discovering semantic relations between the question and the answer. As an example, (Harabagiu and Maiorano, 1999) describes answer validation as an abductive inference process, where an answer is valid with respect to a question if an explanation for it, based on background knowledge, can be found. Although theoretically well motivated, the use of semantic techniques on open domain tasks is quite expensive both in terms of the involved linguistic resources and in terms of computational complexity, thus motivating a research on alternative solutions to the problem.

This paper presents a novel approach to answer validation based on the intuition that the amount of implicit knowledge which connects an answer to a question can be quantitatively estimated by exploiting the redundancy of Web information. The hypothesis is that the number of documents that can be retrieved from the Web in which the question and the answer co-occur can be considered a significant clue of the validity of the answer. Documents are searched in the Web by means of *validation patterns*, which are derived from a linguistic processing of the question and the answer. In order to test this idea a system for automatic answer validation has been implemented and a number of experiments have been carried out on questions and answers provided by the TREC-2001 participants. The advantages of this approach are its simplicity on the one hand and its efficiency on the other.

Automatic techniques for answer validation are of great interest for the development of open domain QA systems. The availability of a completely automatic evaluation procedure makes it feasible QA systems based on generate and test approaches. In this way, until a given answer is automatically

Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 425-432.

proved to be correct for a question, the system will carry out different refinements of its searching criteria checking the relevance of new candidate answers. In addition, given that most of the QA systems rely on complex architectures and the evaluation of their performances requires a huge amount of work, the automatic assessment of the relevance of an answer with respect to a given question will speed up both algorithm refinement and testing.

The paper is organized as follows. Section 2 presents the main features of the approach. Section 3 describes how validation patterns are extracted from a question-answer pair by means of specific question answering techniques. Section 4 explains the basic algorithm for estimating the answer validity score. Section 5 gives the results of a number of experiments and discusses them. Finally, Section 6 puts our approach in the context of related works.

2 Overall Methodology

Given a question q and a candidate answer a the answer validation task is defined as the capability to assess the relevance of a with respect to q. We assume open domain questions and that both answers and questions are texts composed of few tokens (usually less than 100). This is compatible with the TREC-2001 data, that will be used as examples throughout this paper. We also assume the availability of the Web, considered to be the largest open domain text corpus containing information about almost all the different areas of the human knowledge.

The intuition underlying our approach to answer validation is that, given a question-answer pair ([q,a]), it is possible to formulate 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: we will attack this problem defining the more flexible idea of a *validation pattern*. Second, we have to design an effective and efficient way to check the reliability of a validation pattern: our solution relies on a procedure based on a statistical count of Web searches.

Answers may occur in text passages with low similarity with respect to the question. Passages telling facts may use different syntactic constructions, sometimes are spread in more than one sentence, may reflect opinions and personal attitudes, and often use ellipsis and anaphora. For instance, if the validation statement is "The capital of USA is Washington", we have Web documents containing passages like those reported in Table 1, which can not be found with a simple search of the statement, but that nevertheless contain a significant amount of knowledge about the relations between the question and the answer. We will refer to these text fragments as *validation fragments*.

1. Capital Region USA: Fly-Drive Holidays in and Around Washington D.C.
2. the Insider's Guide to the Capital Area Music Scene (Washington D.C., USA).
3. The Capital Tangueros (Washington, DC Area, USA)
4. I live in the Nation's Capital, Washington Metropolitan Area (USA).
5. in 1790 Capital (also USA's capital): Washington D.C. Area: 179 square km

Table 1: Web search for validation fragments

A common feature in the above examples is the co-occurrence of a certain subset of words (*i.e.* "capital","USA" and "Washington"). We will make use of *validation patterns* that cover a larger portion of text fragments, including those lexically similar to the question and the answer (*e.g.* fragments 4 and 5 in Table 1) and also those that are not similar (*e.g.* fragment 2 in Table 1). In the case of our example a set of validation statements can be generalized by the validation pattern:

[capital <*text*> USA <*text*> Washington]

where <*text*> is a place holder for any portion of text with a fixed maximal length.

To check the correctness of a with respect to q we propose a procedure that measures the number of occurrences on the Web of a validation pattern derived from a and q. A useful feature of such patterns is that when we search for them on the Web they usually produce many hits, thus making statistical approaches applicable. In contrast, searching for strict validation statements generally results in a small number of documents (if any) and makes statistical methods irrelevant. A number of techniques used for finding collocations and co-occurrences of words, such as mutual information, may well be used to search co-occurrence tendency between the question and the candidate answer in the Web. If we verify that such tendency is statistically significant we may consider the validation pattern as consistent and therefore we may assume a high level of correlation between the question and the candidate answer.

Starting from the above considerations and given a question-answer pair $[q, a]$, we propose an answer validation procedure based on the following 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 compute the validation pattern for the pair $[q, a]$;

3. Submit the patterns to the Web and estimate an *answer validity score* considering the number of retrieved documents.

3 Extracting Validation Patterns

In our approach a validation pattern consists of two components: a question sub-pattern (*Qsp*) and an answer sub-pattern (*Asp*).

Building the *Qsp*. A *Qsp* is derived from the input question cutting off non-content words with a stop-words filter. The remaining words are expanded with both synonyms and morphological forms in order to maximize the recall of retrieved documents. Synonyms are automatically extracted from the most frequent sense of the word in WordNet (Fellbaum, 1998), which considerably reduces the risk of adding disturbing elements. As for morphology, verbs are expanded with all their tense forms (*i.e.* present, present continuous, past tense and past participle). Synonyms and morphological forms are added to the *Qsp* and composed in an `OR` clause.

The following example illustrates how the *Qsp* is constructed. Given the TREC-2001 question "When did Elvis Presley die?", the stop-words filter removes "When" and "did" from the input. Then synonyms of the first sense of "die" (*i.e.* "decease", "perish", etc.) are extracted from WordNet. Finally, morphological forms for all the corresponding verb tenses are added to the *Qsp*. The resultant *Qsp* will be the following:

[Elvis <*text*> Presley <*text*> (die `OR` died `OR` dying `OR` perish `OR` ...)]

Building the *Asp*. An *Asp* is constructed in two steps. First, the *answer type* of the question is identified considering both morpho-syntactic (a part of speech tagger is used to process the question) and semantic features (by means of semantic predicates defined on the WordNet taxonomy; see (Magnini et al., 2001) for details). Possible answer types are: DATE, MEASURE, PERSON, LOCATION, ORGANIZATION, DEFINITION and GENERIC. DEFINITION is the answer type peculiar to questions like "What is an atom?" which represent a considerable part (around 25%) of the TREC-2001 corpus. The answer type GENERIC is used for non definition questions asking for entities that can not be classified as named entities (*e.g.* the questions: "Material called linen is made from what plant?" or "What mineral helps prevent osteoporosis?")

In the second step, a rule-based named entities recognition module identifies in the answer string all the named entities matching the answer type category. If the category corresponds to a named entity, an *Asp* for each selected named entity is created. If the answer type category is either DEFINITION or GENERIC, the entire answer string except the stop-words is considered. In addition, in order to maximize the recall of retrieved documents, the *Asp* is expanded with verb tenses. The following example shows how the *Asp* is created. Given the TREC question "When did Elvis Presley die?" and

the candidate answer "though died in 1977 of course some fans maintain", since the answer type category is DATE the named entities recognition module will select [1977] as an answer sub-pattern.

4 Estimating Answer Validity

The answer validation algorithm queries the Web with the patterns created from the question and answer and after that estimates the consistency of the patterns.

4.1 Querying the Web

We use a Web-mining algorithm that considers the number of pages retrieved by the search engine. In contrast, qualitative approaches to Web mining (*e.g.* (Brill et al., 2001)) analyze the document content, as a result considering only a relatively small number of pages. For information retrieval we used the AltaVista search engine. Its advanced syntax allows the use of operators that implement the idea of validation patterns introduced in Section 2. Queries are composed using NEAR, OR and AND boolean operators. The NEAR operator searches pages where two words appear in a distance of no more than 10 tokens: it is used to put together the question and the answer sub-patterns in a single validation pattern. The OR operator introduces variations in the word order and verb forms. Finally, the AND operator is used as an alternative to NEAR, allowing more distance among pattern elements.

If the question sub-pattern Qsp does not return any document or returns less than a certain threshold (experimentally set to 7) the question pattern is relaxed by cutting one word; in this way a new query is formulated and submitted to the search engine. This is repeated until no more words can be cut or the returned number of documents becomes higher than the threshold. Pattern relaxation is performed using word-ignoring rules in a specified order. Such rules, for instance, ignore the focus of the question, because it is unlikely that it occurs in a validation fragment; ignore adverbs and adjectives, because are less significant; ignore nouns belonging to the WordNet classes "abstraction", "psychological feature" or "group", because usually they specify finer details and human attitudes. Names, numbers and measures are preferred over all the lower-case words and are cut last.

4.2 Estimating pattern consistency

The Web-mining module submits three searches to the search engine: the sub-patterns [*Qsp*] and [*Asp*] and the validation pattern [*QAp*], this last built as the composition [*Qsp* NEAR Asp]. The search engine returns respectively: $hits(Qsp)$, $hits(Asp)$ and $hits(Qsp\ \texttt{NEAR}\ Asp)$. The probability $P(A)$ of a pattern A in the Web is calculated by:

$$P(A) = \frac{hits(A)}{MaxPages}$$

where $hits(A)$ is the number of pages in the Web where A appears and $MaxPages$ is the maximum number of pages that can be returned by the search engine. We set this constant experimentally. However in two of the formulas we use (*i.e.* Pointwise Mutual Information and Corrected Conditional Probability) $MaxPages$ may be ignored.

The joint probability P(*Qsp,Asp*) is calculated by means of the validation pattern probability:

$$P(QAp) = P(Qsp\ \texttt{NEAR}\ Asp)$$

We have tested three alternative measures to estimate the degree of relevance of Web searches: Pointwise Mutual Information, Maximal Likelihood Ratio and Corrected Conditional Probability, a variant of Conditional Probability which considers the asymmetry of the question-answer relation. Each measure provides an answer validity score: high values are interpreted as strong evidence that the validation pattern is consistent. This is a clue to the fact that the Web pages where this pattern appears contain validation fragments, which imply answer accuracy.

Pointwise Mutual Information (PMI) (Manning and Schütze, 1999) has been widely used to find co-occurrence in large corpora.

$$PMI(Qsp,Asp) = \frac{P(Qsp,Asp)}{P(Qsp) * P(Asp)}$$

PMI(Qsp,Asp) is used as a clue to the internal coherence of the question-answer validation pattern *QAp*. Substituting the probabilities in the PMI formula with the previously introduced Web statistics, we obtain:

$$\frac{hits(Qsp\ \texttt{NEAR}\ Asp)}{hits(Qsp) * hits(Asp)} * MaxPages$$

Maximal Likelihood Ratio (MLHR) is also used for word co-occurrence mining (Dunning, 1993). We decided to check MLHR for answer validation because it is supposed to outperform PMI in case of sparse data, a situation that may happen in case of questions with complex patterns that return small number of hits.

$$MLHR(Qsp, Asp) = -2 \log \lambda$$

$$\lambda = \frac{L(p, k_1, n_1)L(p, k_2, n_2)}{L(p_1, k_1, n_1)L(p_2, k_2, n_2)}$$

where $L(p, k, n) = p^k(1-p)^{n-k}$
$p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$
$p = \frac{k_1+k_2}{n_1+n_2}$
$k_1 = hits(Qsp, Asp)$, $k_2 = hits(Qsp, -Asp)$
$n_1 = hits(Asp)$, $n_2 = hits(-Asp)$

Here $hits(Qsp, -Asp)$ is the number of appearances of *Qsp* when *Asp* is not present and it is calculated as $hits(Qsp) - hits(Qsp\ \texttt{NEAR}\ Asp)$.

Similarly, $hits(-Asp)$ is the number of Web pages where *Asp* does not appear and it is calculated as $MaxPages - Asp$.

Corrected Conditional Probability (CCP) in contrast with PMI and MLHR, CCP is not symmetric (*e.g.* generally $CCP(Qsp, Asp) \neq CCP(Asp, Qsp)$). This is based on the fact that we search for the occurrence of the answer pattern Asp only in the cases when *Qsp* is present. The statistical evidence for this can be measured through $P(Asp|Qsp)$, however this value is corrected with $P(Asp)^{2/3}$ in the denominator, to avoid the cases when high-frequency words and patterns are taken as relevant answers.

$$CCP(Qsp, Asp) = \frac{P(Asp|Qsp)}{P(Asp)^{2/3}}$$

For CCP we obtain:

$$\frac{hits(Qsp\ \texttt{NEAR}\ Asp)}{hits(Qsp) * hits(Asp)^{2/3}} * MaxPages^{2/3}$$

4.3 An example

Consider an example taken from the question answer corpus of the main task of TREC-2001: "Which river in US is known as Big Muddy?". The question keywords are: "river", "US", "known", "Big", "Muddy". The search of the pattern [river NEAR US NEAR (known OR know OR...) NEAR Big NEAR Muddy] returns 0 pages, so the algorithm relaxes the pattern by cutting the initial noun "river", according to the heuristic for discarding a noun if it is the first keyword of the question. The second pattern [US NEAR (known OR know OR...) NEAR Big NEAR Muddy] also returns 0 pages, so we apply the heuristic for ignoring verbs like "know", "call" and abstract nouns like "name". The third pattern [US NEAR Big NEAR Muddy] returns 28 pages, which is over the experimentally set threshold of seven pages.

One of the 50 byte candidate answers from the TREC-2001 answer collection is "recover Mississippi River". Taking into account the answer type LOCATION, the algorithm considers only the named entity: "Mississippi River". To calculate answer validity score (in this example PMI) for [Mississippi River], the procedure constructs the validation pattern: [US NEAR Big NEAR Muddy NEAR Mississippi River] with the answer sub-pattern [Mississippi River]. These two patterns are passed to the search engine, and the returned numbers of pages are substituted in the mutual information expression at the places of $hits(Qsp\ \texttt{NEAR}\ Asp)$ and $hits(Asp)$ respectively; the previously obtained number (*i.e.* 28) is substituted at the place of $hits(Qsp)$. In this way an answer validity score of 55.5 is calculated. It turns out that this value is the maximal validity score for all the answers of this question. Other correct answers from the TREC-2001 collection contain as name entity "Mississippi". Their answer validity score is 11.8, which is greater than 1.2 and also greater than $0.2 * Maximal\_Validity\_Score$ $(= 11.1)$. This score (*i.e.* 11.8) classifies them as relevant answers. On the other hand, all the wrong answers has validity score below 1 and as a result all of them are classified as irrelevant answer candidates.

5 Experiments and Discussion

A number of experiments have been carried out in order to check the validity of the proposed answer validation technique. As a data set, the 492 questions of the TREC-2001 database have been used. For each question, at most three correct answers and three wrong answers have been randomly selected from the TREC-2001 participants' submissions, resulting in a corpus of 2726 question-answer pairs (some question have less than three positive answers in the corpus). As said before, AltaVista was used as search engine.

A baseline for the answer validation experiment was defined by considering how often an answer occurs in the top 10 documents among those (1000 for each question) provided by NIST to TREC-2001 participants. An answer was judged correct for a question if it appears at least one time in the first 10 documents retrieved for that question, otherwise it was judged not correct. Baseline results are reported in Table 2.

We carried out several experiments in order to check a number of working hypotheses. Three independent factors were considered:

Estimation method. We have implemented three measures (reported in Section 4.2) to estimate an answer validity score: PMI, MLHR and CCP.

Threshold. We wanted to estimate the role of two different kinds of thresholds for the assessment of answer validation. In the case of an *absolute threshold*, if the answer validity score for a candidate answer is below the threshold, the answer is considered wrong, otherwise it is accepted as relevant. In a second type of experiment, for every question and its corresponding answers the program chooses the answer with the highest validity score and calculates a *relative threshold* on that basis (*i.e.* $threshold = k * Max\_Validity\_score$). However the relative threshold should be larger than a certain minimum value.

Question type. We wanted to check performance variation based on different types of TREC-2001 questions. In particular, we have separated definition and generic questions from true named entities questions.

Tables 2 and 3 report the results of the automatic answer validation experiments obtained respectively on all the TREC-2001 questions and on the subset of definition and generic questions. For each estimation method we report precision, recall and success rate. Success rate best represents the performance of the system, being the percent of $[q, a]$ pairs where the result given by the system is the same as the TREC judges' opinion. Precision is the percent of $[q, a]$ pairs estimated by the algorithm as relevant, for which the opinion of TREC judges was the same. Recall shows the percent of the relevant answers which the system also evaluates as relevant.

	P (%)	*R (%)*	*SR (%)*
Baseline	50.86	4.49	52.99
CCP - rel.	77.85	82.60	81.25
CCP - abs.	74.12	81.31	78.42
PMI - rel.	77.40	78.27	79.56
PMI - abs.	70.95	87.17	77.79
MLHR - rel.	81.23	72.40	79.60
MLHR - abs.	72.80	80.80	77.40

Table 2: Results on all 492 TREC-2001 questions

	P (%)	*R (%)*	*SR (%)*
CCP - rel.	85.12	84.27	86.38
CCP - abs.	83.07	78.81	83.35
PMI - rel.	83.78	82.12	84.90
PMI - abs.	79.56	84.44	83.35
MLHR - rel.	90.65	72.75	84.44
MLHR - abs.	87.20	67.20	82.10

Table 3: Results on 249 named entity questions

The best results on the 492 questions corpus (CCP measure with relative threshold) show a success rate of 81.25%, *i.e.* in 81.25% of the pairs the system evaluation corresponds to the human evaluation, and confirms the initial working hypotheses. This is 28% above the baseline success rate. Precision and recall are respectively 20-30% and 68-87% above the baseline values. These results demonstrate that the intuition behind the approach is motivated and that the algorithm provides a workable solution for answer validation.

The experiments show that the average difference

between the success rates obtained for the named entity questions (Table 3) and the full TREC-2001 question set (Table 2) is 5.1%. This means that our approach performs better when the answer entities are well specified.

Another conclusion is that the relative threshold demonstrates superiority over the absolute threshold in both test sets (average 2.3%). However if the percent of the right answers in the answer set is lower, then the efficiency of this approach may decrease.

The best results in both question sets are obtained by applying CCP. Such non-symmetric formulas might turn out to be more applicable in general. As conditional corrected (CCP) is not a classical co-occurrence measure like PMI and MLHR, we may consider its high performance as proof for the difference between our task and classic co-occurrence mining. Another indication for this is the fact that MLHR and PMI performances are comparable, however in the case of classic co-occurrence search, MLHR should show much better success rate. It seems that we have to develop other measures specific for the question-answer co-occurrence mining.

6 Related Work

Although there is some recent work addressing the evaluation of QA systems, it seems that the idea of using a fully automatic approach to answer validation has still not been explored. For instance, the approach presented in (Breck et al., 2000) is semi-automatic. The proposed methodology for answer validation relies on computing the overlapping between the system response to a question and the stemmed content words of an answer key. All the answer keys corresponding to the 198 TREC-8 questions have been manually constructed by human annotators using the TREC corpus and external resources like the Web.

The idea of using the Web as a corpus is an emerging topic of interest among the computational linguists community. The TREC-2001 QA track demonstrated that Web redundancy can be exploited at different levels in the process of finding answers to natural language questions. Several studies (*e.g.* (Clarke et al., 2001) (Brill et al., 2001)) suggest that the application of Web search can improve the precision of a QA system by 25-30%. A common feature of these approaches is the use of the Web to introduce data redundancy for a more reliable answer extraction from local text collections. (Radev et al., 2001) suggests a probabilistic algorithm that learns the best query paraphrase of a question searching the Web. Other approaches suggest training a question-answering system on the Web (Mann, 2001).

The Web-mining algorithm presented in this paper is similar to the PMI-IR (Pointwise Mutual Information - Information Retrieval) described in (Turney, 2001). Turney uses PMI and Web retrieval to decide which word in a list of candidates is the best synonym with respect to a target word. However, the answer validity task poses different peculiarities. We search how the occurrence of the question words influence the appearance of answer words. Therefore, we introduce additional linguistic techniques for pattern and query formulation, such as keyword extraction, answer type extraction, named entities recognition and pattern relaxation.

7 Conclusion and Future Work

We have presented a novel approach to answer validation based on the intuition that the amount of implicit knowledge which connects an answer to a question can be quantitatively estimated by exploiting the redundancy of Web information. Results obtained on the TREC-2001 QA corpus correlate well with the human assessment of answers' correctness and confirm that a Web-based algorithm provides a workable solution for answer validation.

Several activities are planned in the near future.

First, the approach we presented is currently based on fixed validation patterns that combine single words extracted both from the question and from the answer. These *word-level* patterns provide a broad coverage (i.e. many documents are typically retrieved) in spite of a low precision (i.e also weak correlations among the keyword are captured). To increase the precision we want to experiment other types of patterns, which combine words into larger units (e.g. phrases or whole sentences). We believe that the answer validation process can be improved both considering pattern variations (from word-level to phrase and sentence-level), and the trade-off between the precision of the search pattern and the

number of retrieved documents. Preliminary experiments confirm the validity of this hypothesis.

Then, a generate and test module based on the validation algorithm presented in this paper will be integrated in the architecture of our QA system under development. In order to exploit the efficiency and the reliability of the algorithm, such system will be designed trying to maximize the recall of retrieved candidate answers. Instead of performing a deep linguistic analysis of these passages, the system will delegate to the evaluation component the selection of the right answer.

References

E.J. Breck, J.D. Burger, L. Ferro, L. Hirschman, D. House, M. Light, and I. Mani. 2000. How to Evaluate Your Question Answering System Every Day and Still Get Real Work Done. In *Proceedings of LREC-2000*, pages 1495–1500, Athens, Greece, 31 May - 2 June.

E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. 2001. Data-Intensive Question Answering. In *TREC-10 Notebook Papers*, Gaithesburg, MD.

C. Clarke, G. Cormack, T. Lynam, C. Li, and G. McLearn. 2001. Web Reinforced Question Answering (MultiText Experiments for TREC 2001). In *TREC-10 Notebook Papers*, Gaithesburg, MD.

T. Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.

C. Fellbaum. 1998. *WordNet, An Electronic Lexical Database*. The MIT Press.

S. Harabagiu and S. Maiorano. 1999. Finding Answers in Large Collections of Texts: Paragraph Indexing + Abductive Inference. In *Proceedings of the AAAI Fall Symposium on Question Answering Systems*, pages 63–71, November.

B. Magnini, M. Negri, R. Prevete, and H. Tanev. 2001. Multilingual Question/Answering: the DIOGENE System. In *TREC-10 Notebook Papers*, Gaithesburg, MD.

G. S. Mann. 2001. A Statistical Method for Short Answer Extraction. In *Proceedings of the ACL-2001 Workshop on Open-Domain Question Answering*, Toulouse, France, July.

C.D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT PRESS, Cambridge,Massachusets.

H. R. Radev, H. Qi, Z. Zheng, S. Blair-Goldensohn, Z. Zhang, W. Fan, and J. Prager. 2001. Mining the Web for Answers to Natural Language Questions. In *Proceedings of 2001 ACM CIKM*, Atlanta, Georgia, USA, November.

M. Subbotin and S. Subbotin. 2001. Patterns of Potential Answer Expressions as Clues to the Right Answers. In *TREC-10 Notebook Papers*, Gaithesburg, MD.

P.D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML2001*, pages 491–502, Freiburg, Germany.

R. Zajac. 2001. Towards Ontological Question Answering. In *Proceedings of the ACL-2001 Workshop on Open-Domain Question Answering*, Toulouse, France, July.